Multimodal Neuroimaging Feature Learning With Multimodal Stacked Deep Polynomial Networks for Diagnosis of Alzheimer's Disease

Jun Shi, Xiao Zheng, Yan Li, Qi Zhang, Shihui Ying, *Member, IEEE*, and ADNI

***Abstract*—The accurate diagnosis of Alzheimer's disease (AD) and its early stage, i.e., mild cognitive impairment, is essential for timely treatment and possible delay of AD. Fusion of multimodal neuroimaging data, such as magnetic resonance imaging (MRI) and positron emission tomography (PET), has shown its effectiveness for AD diagnosis. The deep polynomial networks (DPN) is a recently proposed deep learning algorithm, which performs well on both large-scale and small-size datasets. In this study, a multimodal stacked DPN (MM-SDPN) algorithm, which MM-SDPN consists of two-stage SDPNs, is proposed to fuse and learn feature representation from multimodal neuroimaging data for AD diagnosis. Specifically speaking, two SDPNs are first used to learn high-level features of MRI and PET, respectively, which are then fed to another SDPN to fuse multimodal neuroimaging information. The proposed MM-SDPN algorithm is applied to the ADNI dataset to conduct both binary classification and multiclass classification tasks. Experimental results indicate that MM-SDPN is superior over the state-of-the-art multimodal feature-learning-based algorithms for AD diagnosis.**

***Index Terms*—Alzheimer's disease, deep learning, deep polynomial networks, multimodal stacked deep polynomial networks, multimodal neuroimaging.**

I. Introduction

ALZHEIMER'S DISEASE (AD) is one of the most prevalent progressive neurodegenerative brain disorders, resulting in a gradual, irreversible loss of memory and other cognitive functions in elderly people worldwide [1]. It is expected that the number of individuals with AD will grow to 100 million in 2050 world-wide [2]. Because of the dramatic increase in the prevalence of AD, the accurate diagnosis of AD and its early stage, i.e. mild cognitive impairment (MCI), is critical for timely treatment and possible delay of AD.

Over the past decades, neuroimaging techniques, such as magnetic resonance imaging (MRI), functional MRI (fMRI) and positron emission tomography (PET), have profoundly advanced neuroscience research and clinical application [3], [4]. The identifiable imaging biomarkers have been effectively utilized for the diagnosis or prognosis of AD, due to their advantages of visualization and quantitative measurement of brain structural and functional information by neuroimaging [5]–[7].

In recent years, the neuroimaging based computer-aided diagnosis (CAD) for AD has attracted considerable attention [4], [8]–[10]. Moreover, since different neuroimaging modalities, such as MRI and PET, can provide different and complementary information to improve diagnostic performance for AD [4], [11], [12], the multimodal neuroimaging based AD classification has attracted considerable attention [4], [13]–[18].

The performance of machine learning algorithms in CAD generally depends on data representation (or feature), and therefore feature extraction becomes a critical step in the classification framework [10], [19]. The most commonly used feature extraction methods for neuroimaging data can be roughly divided into four categories [8], [20]: 1) The voxel-based approaches that simply and directly extract features from voxel intensity; 2) The vertex-based approaches whose features are defined at the vertex-level on the cortical surface; 3) The region of interest (ROI) based approaches that extract features from predefined brain regions; 4) patch-based approaches that learn new feature representation from local patches. The voxel- and vertex-based features usually have very high dimensionality, and therefore dimensionality reduction is important to achieve more compact and effective features. The ROI-based features are widely used, because they not only have relatively low feature dimensionality, but also cover the whole brain. However, the features extracted from ROIs are somewhat coarse and cannot reflect small or subtle changes involved in the brain diseases. The patch-based features are learned from the whole brain, and can effectively capture the diseased-related pathologies. As a result, these features usually obtain superior classification results

Manuscript received October 1, 2016; revised February 16, 2016 and January 11, 2017; accepted January 12, 2017. Date of publication January 19, 2017; date of current version January 3, 2018. This work was supported in part by the National Natural Science Foundation of China (61471231, 81627804, 61401267, U1201256, 61471245, 11471208) and in part by the Projects of Guangdong R/D Foundation and the Fundamental Science Projects of Shenzhen City (JCYJ20140418091413514, ZDSYS20140508141148477, JCYJ20160308095019383). *(Corresponding author: Shihui Ying).*

J. Shi, X. Zheng, and Q. Zhang are with the Institute of Biomedical Engineering, School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: junshi@staff.shu.edu.cn; zx222@i.shu.edu.cn; zhangq@shu.edu.cn).

Y. Li is with the Shenzhen City Key Laboratory of Embedded System Design, Shenzhen Laboratory of IC Design for Internet of Things, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: liyan@szu.edu.cn).

S. Ying is with the Department of Mathematics, School of Science, Shanghai University, Shanghai 200444, China (e-mail: shying@shu.edu.cn).

Digital Object Identifier 10.1109/JBHI.2017.2655720

[20]–[24]. The way of patch-based features essentially is a procedure of feature re-representation to learn and then generate a new feature space with learning algorithms, among which deep learning (DL) has achieved the state-of-the-art performance [20], [24].

DL has achieved great success in various applications, since it was first introduced by Hinton *et al.* in 2006 [25], [26]. In contrast to the conventional shallow-structured learning architectures, DL develops a layered, hierarchical architecture to yield high-level and more effective data representation [19], [27], [28]. In recent years, DL gains its good reputation in the domain of medical imaging, such as medical image classification, detection and segmentation [29]–[35].

In the case of neuroimaging data, DL can efficiently discover latent or hidden representation, and therefore it has been successfully applied to diagnosis of AD and other brain diseases. Suk *et al.* used the multimodal deep restricted Boltzmann machine (RBM) to learn features from huge 3D patches in multimodal neuroimaging data for AD/MCI diagnosis [20]. Gupta *et al.* used stacked autoencoder (SAE) to learn a set of bases from natural images and then applied convolution network to get more effective feature representation for AD classification [23]. Brosch *et al.* learned a low-dimensional manifold of brain volumes by the deep belief networks (DBN) algorithm to detect the modes of variations correlating to demographic and disease parameters for AD [36]. Sergey *et al.* also used RBM and DBN to learn features from MRI and fMRI for schizophrenia diagnosis [34]. *Ithaput et al.* developed a randomized denoising autoencoders algorithm to learn representation from high-dimensional neuroimaging features for AD clinical trials [37]. Liu *et al.* also proposed an SAE-based multimodal neuroimaging feature learning algorithm to learn feature representation from ROI-based features for AD diagnosis [38]. Payan and Montana combined sparse autoencoder and convolution neural network to learn feature representation from local patches for AD prediction [24]. Suk and Shen integrated SAE with the multitask feature selection and multi-kernel learning (MKL) algorithms to learn latent feature representation from ROI features of multimodal neuroimaging data and cerebrospinal fluid (CSF) features for AD classification [39]. They also proposed a deep architecture to select features with the sparse multi-task learning in a hierarchical fashion for AD diagnosis [40]. Li *et al.* developed a dropout technique based robust multitask deep learning framework to improve the representation of ROI features for AD/MCI diagnosis [41]. Chen *et al.* proposed to extract features from the longitudinal T1 MRI images, and then apply the stacked denoising sparse autoencoders (SDSAE) to fuse these features for AD staging analysis [42].

On the other hand, the deep polynomial networks (DPN) is a newly developed supervised DL algorithm, in which each node calculates a linear or quadratic function of its inputs, and hence the learned predictors are the polynomial functions over input space [43]. It doesn't rely on complicated heuristics, and is easy to be implemented. DPN has also achieved competitive performance as compared with DBN algorithm on some commonly used large-scale image datasets [43]. It is also worth noting that DPN can also compactly represent any function on a finite sample dataset, because its algorithm structure is originally developed for small dataset [43].

Since the neuroimaging dataset generally has small labeled samples [10], as an appropriate feature representation algorithm for the small dataset, the supervised DPN will potentially learn superior feature representation from small neuroimaging data for AD diagnosis. On the other hand, the layer-wise stacking of feature extraction often yields better representations in DL [19], such as DBN and SAE [25], [44], which motivates to develop a stacked DPN (SDPN) algorithm to learn higher-level feature representation. Moreover, it has been proved that the multimodal DL algorithms, which can simultaneously learn and fuse multimodal neuroimaging data, outperform the single-modal DL algorithms for AD classification [20], [38], [39], [41]. Therefore, it is worth investigating a multimodal stacked DPN algorithm.

In this work, a multimodal stacked DPN (MM-SDPN) algorithm is proposed motivated by the above-mentioned factors, and then it is applied to the multimodal neuroimaging based AD diagnosis. MM-SDPN can effectively fuse and learning feature representation from multimodal neuroimaging.

The rest of this paper is organized as follows. Section II provides the introduction about the original DPN and the proposed MM-SDPN algorithms. The conducted experiments and the results are given in Section III to evaluate the performance of the proposed MM-SDPN algorithm. The discussion and conclusion are presented in Sections IV and V, respectively.

II. Methods

A. Deep Polynomial Networks Algorithm

DPN algorithm is devoted to learn polynomial predictors by a deep network architecture that provides a good approximate basis for the values attained by polynomials over the training samples. Below we give a brief introduction about DPN algorithm, and for more details we refer to [43].

Fig. 1 shows a architecture of DPN with four-layer networks as an example.

Let $\{(x_i, y_i)\}_{i=1}^{m}$ be m training samples, where x_i is a d-dimension sample and y_i is the corresponding label value. The multivariate polynomials on $\mathbf{R}^d$ are defined by

$$p\ (x) = \sum_{i=0}^{\Delta} \sum_{\alpha^i} w_{\alpha^i} \prod_{l=1}^{d} x_l^{\alpha_l^i} \tag{1}$$

where Δ is the degree of the polynomial, i is the index of degree, α^i is the d-dimensional index with $\sum_{l=1}^{d} \alpha_l^i = i$, and w_{α^i} is the weight at the index of α^i. In DPN, we use $n_j^i(\cdot)$ to describe the j-th node in the i-th layer, as a function of its inputs. For simplification, the function calculated in each node is either a linear one or a weighted product of two inputs.

When constructing the first layer network in DPN, the set of values by degree-1 polynomials (linear) function over training samples is given by

$$\{(\mathrm{w}, [1\ \mathrm{x1}],\ \ldots,\ \mathrm{w}, [1\ \mathrm{xm}]) : \mathrm{w} \in \mathrm{Rd} + 1\} \tag{2}$$

which is the $(d+1)$-dimensional linear subspace of $\mathbf{R}^m$. Here $\langle \cdot, \cdot \rangle$ is the inner product. A basis can be conducted for this

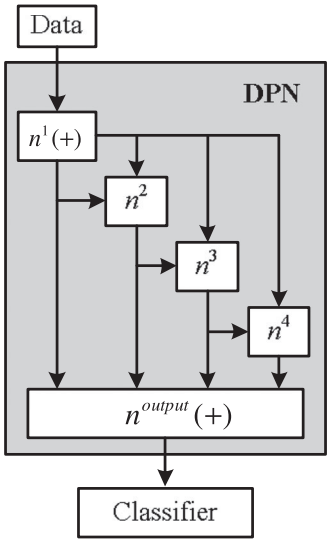

Fig. 1. Schematic diagram of the network's architecture in a degree-4 DPN. Each n^l represents a layer of nodes. (+) means a layer of nodes, which calculates functions of the form $\mathrm{n}(z) = \sum_i w_i z_i$, while other layers consist of nodes that compute functions of the form $\mathrm{n}(z) = n((z_{i(1)}, z_{i(2)})) = wz_{i(1)}z_{i(2)}$.

linear independent set by singular value decomposition (SVD) algorithm. To simplify the model, we denote by a matrix W which maps [**1 X**] into the constructed basis, where **1** is the all-ones vector, and **X** is a matrix of samples. The columns of W mean the d + 1 linear functions forming the first layer in DPN; That is, for all $j = 1, \ldots, d+1$, the *j*-th node of first layer is the function

$$n_j^1(x) = \mathrm{Wj}, [1\ \mathrm{X}] \tag{3}$$

Here, $\{(n_j^1(x_1), \ldots, n_j^1(x_m))\}_{j=1}^{d+1}$ is a basis for all value obtained by degree-1 polynomials over training samples. Let F^1 be an $m \times (d+1)$ matrix with $F_{i,j}^1 = n_j^1(x_i)$. Up to now, a one-layer network is built, whose outputs span all values obtained by linear functions on the training samples.

By the decomposition theorem of polynomials in [43], any degree-2 polynomial can be written as

$$\sum_i \left(\sum_j \alpha_i^{(g_i)} n_j^1(x)\right)\left(\sum_j \alpha_s^{(h_i)} n_s^1(x)\right) + \left(\sum_j \alpha_j^{(k)} n_j^1(x)\right)$$
$$= \sum_{j,i} n_j^1(x)\, n_s^1(x) \left(\sum_i \alpha_j^{(g_i)} \alpha_s^{(h_i)}\right) + \sum_j n_j^1(x) \left(\alpha_j^{(k)}\right) \tag{4}$$

where α is a scalar, and the superscripts $\mathrm{g_i}$ and $\mathrm{h_i}$ represent the degree 1 polynomials with $g_i(x) = \sum_j \alpha_i^{(g_i)} n_j^1(x)$ and $h_i(x) = \sum_j \alpha_s^{(h_i)} n_s^1(x)$, respectively. It indicates that the vector of values obtained by any degree-2 polynomial spans the vector of values by nodes in the first layer, and products of the outputs of every two nodes in the first layer.

Furthermore, we define $\tilde{F}^2$ by

$$\tilde{\mathrm{F}}^2 = \left[\left(\mathrm{F}_1^1 \bigcirc \mathrm{F}_1^1\right) \cdots \left(\mathrm{F}_1^1 \bigcirc \mathrm{F}_{|\mathrm{F}_1|}^1\right) \cdots \left(\mathrm{F}_{|\mathrm{F}_1|}^1 \bigcirc \mathrm{F}_1^1\right) \cdots \left(\mathrm{F}_{|\mathrm{F}_1|}^1 \bigcirc \mathrm{F}_{|\mathrm{F}_1|}^1\right)\right] \tag{5}$$

where $\circ$ denotes the Hadamard product, $|F_1|$ is the number of the columns of matrix F_1, *and* F_i *means the i-th column vector of the matrix F*. Then, the concatenated new matrix $[F\ \tilde{F}^2]$ is in the span of all possible values attainable by degree-2 polynomials. SVD is again performed to construct the basis. Let F^2 be a subset of the columns of $\tilde{F}^2$. F^2 generates the basis of degree-2 polynomial, and it can be obtained by SVD method to select the linear independent columns from $\tilde{F}^2$. The columns of F^2 then specify the second layer network. Here, each such column represents $F_i^1 \circ F_j^1$, which corresponds in turn to a node in the second layer that calculates the product of nodes $n_i^1(\cdot)$ and $n_j^1(\cdot)$ in the first layer.

For a simple notation, F is redefined as an augmented matrix $[F\ F^t]$. Then, at each iteration t, matrix F is maintained, whose columns form a basis for the values obtained by all polynomials of degree $\leq$ (t-1). The new matrix can be represented as

$$\tilde{F}^t = \left[\left(F_1^{t-1} \bigcirc F_1^1\right) \cdots \left(F_1^{t-1} \bigcirc F_{|F_1|}^1\right) \cdots \left(F_{|F_1|}^{t-1} \bigcirc F_1^1\right) \cdots \left(F_{|F^{t-1}|}^{t-1} \bigcirc F_1^1\right)\right] \tag{6}$$

where the columns of $[F\ F^t]$ form a basis for the columns of $[F\ \tilde{F}^t]$. By adding this newly built layer, a network is constructed, whose outputs form a basis for the values obtained by all polynomials of degree $\leq t$ over the training samples. $\tilde{F}_t$ is then transformed to F^t via a matrix W of size $|F^{t-1}| \times |F^1|$ to maintain numerical stability, which is given by

$$F_s^t := W_{i(s),j(s)}\ F_{i(s)}^{t-1} \bigcirc F_{j(j)}^1\ s = 1, 2, \ldots, |F^t| \tag{7}$$

where $i(s)$ indicate that i is exactly a function of s, and W is a projection matrix, which maps $\tilde{F}^t$ onto the basis F^t of degree t polynomials and is gained by Gram-Schmidt procedure or more stable alternative methods. This procedure is same with constructing W in the first layer.

A matrix F is finished after $\Delta - 1$ iterations, whose columns form a basis for all values attained by polynomials of degree $\leq (\Delta - 1)$ over the training samples. The above procedures to build network in DPN are shown in Fig. 2. It can be found that DPN has a feedforward architecture, which makes DPN very simple.

On top of this network, the learned features from the feature output layer $n^{\mathrm{output}}(+)$ are fed to a standard L_2-regularized hinge loss based simple linear classifier as the final decision output, which is solved by the stochastic gradient descent optimization on the hinge loss function [43]. Of course, other classifiers can also be used for DPN.

DPN is now constructed by above-mentioned method. However, the present algorithm can only be used for small datasets with limited nodes. When applying DPN to large training samples, the network width (i.e. the number of nodes created in each layer) also becomes very large, resulting in huge networks

Initialize F as an empty matrix;

Compute SVD to find a W

Create first layer:

$\forall i \in \{1, \ldots |F^1|\}, n_i^1(x) \coloneqq <W_i, [\mathbf{1}\ \mathbf{X}]>$,

$F^1 = n_i^1(x)$; //Columns of F^1 are linearly independent

Constructing 2, 3, … layer:

If error sufficiently small, **break**

$$\tilde{F}^t = \left[(F_1^{t-1} \circ F_1^1)(F_1^{t-1} \circ F_2^1) \cdots (F_{|F^{t-1}|}^{t-1} \circ F_{|F^1|}^1)\right];$$

$$(F^t, W^t) \coloneqq \mathbf{BuildBasis^t}(F, \tilde{F}^t)$$

//Columns of $[F, \tilde{F}^t]$ are linearly independent;

//W^t is such that $F_s^t = W_{i(s),j(s)} F_{i(s)}^{t-1} \circ F_{j(j)}^1$

F:=$[F\ F^t]$

(a)

Output Layer (F)

$w \coloneqq arg\ min_{w \in R^{|F|}} l\ (Fw, y)$

$n^{output}(\cdot) \coloneqq \langle \boldsymbol{w}, n(\cdot) \rangle$,

where $n(\cdot) = (n_1^1(\cdot), n_2^1(\cdot), \cdots, n_{|F^{t-1}|}^{t-1}(\cdot))$

consists of the outputs of all nodes in network

Let error be the error of $n^{output}(\cdot)$ on validation dataset

Return $(n^{output}(\cdot),$ error)

(b)

Fig. 2. (a) The main algorithm of DPN; (b) The output layer construction procedure.

with high computational complexity. To address this issue, a modified solution is proposed to explicitly constrain the network width in each iteration, which uses a smaller partial basis to generate small nodes in a layer [43]. Particularly, in DPN, we can give up on exactly spanning $\tilde{F}^t$, but seek to "approximately span" it by using a smaller partial basis of bounded size r, resulting in a layer of width r. Further, a network is created with sparse connections where most nodes depend only on the input of other two nodes, which makes the network computationally fast. Therefore, this new solution combined with the sparse connection of nodes makes DPN even work well for large-scale data, without losing the advantage of effectiveness and low computational complexity for small data.

For a supervised DPN, the first layer network computes a linear transformation that transforms the augmented data matrix $[\mathbf{1}\ \mathbf{X}]$ into its top k leading singular vectors by principal component analysis (PCA). The next layer networks will adopt a stand standard orthogonal least squares (OLS) procedure to greedily pick the columns of $\tilde{F}^t$ mostly relevant for prediction. This OLS in fact involves a supervised approach, in which we iteratively pick the column of $\tilde{F}^t$ whose residual (after projecting on the existing basis F) is most correlated with the residual of prior labels.

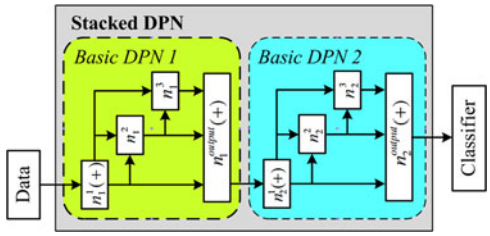

Fig. 3. Schematic diagram of the network architecture of a SDPN with two-level basic DPN blacks.

B. Stacked Deep Polynomial Networks

Hinton *et al.* argue that in DBN, the representation can be improved by giving the output of one RBM as input to another RBM, and the posterior RBM then provide a more complex representation of the input [25]. To deal with this issue, a stacked DPN (SDPN) algorithm is proposed in this work by inheriting the stacking method used in DL.

In SDPN, multiple basic DPNs can be stacked upon each other to form a deep hierarchy, in which the output of a basic DPN is wired to the input of next basic DPN at successive level. Fig. 3 shows an example of two-level SDPN model, which can be extended to m-level.

For an m-level SDPN, the original feature vector is input to the first-level basic DPN block with n-layer networks to generate the learned feature output $n_1^{\text{output}}(+)$, which will be used as the input of the next level basic DPN. Once the current ith-level basic DPN finishes its training, the output $n_i^{\text{output}}(+)$ will be fed to the subsequent level of basic DPN for training. The basic DPN blocks, each consisting of a simple and easy-to-learn module, are stacked to form the overall deep networks. Each basic DPN is trained in a supervised, block-wise fashion, without the need for back propagation, which is different from other popular deep architectures [19]. Therefore, SDPN is very simple with relatively low computational complexity compared with other DL algorithms with back propagation strategy.

It is worth noting that the first layer network of each basic DPN will still calculate a linear transformation that transforms the augmented data matrix $[\mathbf{1}\ \mathbf{X}]$ into its top k leading singular vectors by PCA [43].

C. Multimodal Stacked DPN

In order to fuse and learn feature representation from multimodal data by SDPN, a simple solution is to concatenate all vectors of different modalities. However, this simple concatenation strategy ignores the diversity of multiple modalities to a certain extent, and it cannot well explore the complementary nature and represent the highly nonlinear correlations among multiple modalities. Therefore, we propose a MM-SDPN algorithm based on a two-stage SDPN as shown in Fig. 4.

In the first stage, every neuroimaging data will be fed to its corresponding SDPN module to learn high-level feature representation. The high-level features in each specific modality reflect its own attribute, but without correlative information among different modalities. All the learned features are then

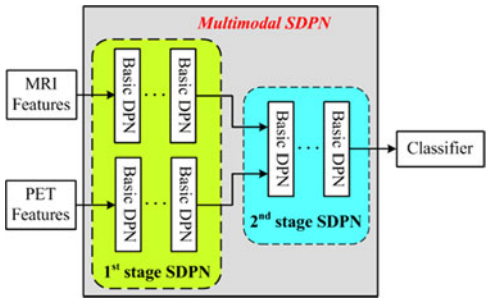

Fig. 4. Flowchart of a MM-SDPN based AD classification with MRI and PET features.

fed to a new SDPN module in the second stage so as to associate all modalities. As a result, the final learned higher-level features contain both the intrinsic properties of each modality and the correlations among all modalities. Therefore, the features learned by SDPN are more discriminative and robust.

In this work, we adopt MRI and PET as two neuroimaging modalities, which are commonly used in multimodal neuroimaging based AD classification [13]–[18], [38]. Notably, the fusion strategy in MM-SDPN is different from that in a multimodal SAE based algorithm in [38] to discover the synergy between MRI and PET. In [38], Liu *et al.* adopt the pre-training method with a proportion of corrupted inputs, which had only one modality presented. Particularly, they randomly hide one modality by setting these inputs as 0, and then present the rest of the training samples with both modalities. The hidden layer in the first-level AE is trained to reconstruct all of the original inputs from the inputs mixed with hidden modalities. Both the original inputs and the corrupted inputs are propagated to the higher network layers independently to achieve the clean representation. While in our MM-SDPN algorithm, since DPN performs the feedforward supervised learning in each network layer without fine-turning, it is difficult to conduct the same learning strategy as in [38] to infer the correlations between MRI and PET. Therefore, the shared representation is learned by jointly training a second stage SDPN with the concatenated MRI and PET features learned in the first stage. It is similar to the simple fusion method used in [42].

III. Experiments and Results

A. Neuroimaging Data Preprocessing

To evaluate the performance of proposed MM-SDPN algorithm, the multimodal neuroimaging data from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database are used here (www.loni.ucla.edu/ADNI) [45]. According to reference [14], we used the same MRI and PET images from 51 AD patients, 99 MCI patients (43 MCI converters (MCI-C), who progressed to AD, and 56 MCI non-converters (MCI-NC), who did not progress to AD in 18 months), and 52 normal controls (NC) with same preprocessing and feature generation methods.

Specifically, the preprocessing is first performed on MRI images including anterior commissure (AC)-posterior commissure (PC) correction, intensity inhomogeneity by N3 algorithm [46], and skull-stripping and removal of cerebellum with the algorithm proposed by Wang *et al.* [47], [48]. MR images are then segmented into three different tissues, namely grey matter, white matter, and cerebrospinal fluid by the FAST algorithm in the FSL package [49]. After registration by HAMMER algorithm [50], each MR image is partitioned into 93 ROIs based on a template with 93 manually labeled ROIs by Kabani *et al.* [51]. The volumes of gray matter tissue are then calculated as a feature for each ROI, resulting in 93 features. Each PET image is then aligned to its corresponding MRI image with a rigid registration. The average intensities of the same ROIs are calculated as the features of PET image. Consequently, 93 features are extracted from MRI and PET images, respectively.

B. Performance Evaluation

Four classification tasks are performed, namely AD vs. NC, MCI vs. NC, MCI-C vs. MCI-NC, and AD vs. MCI-C vs. MCI-NC vs. NC.

The proposed MM-SDPN algorithm is first compared with the following original DPN and SDPN algorithms: (1) features learned by the original DPN with 3-layer networks from the original MRI and PET features, respectively (named DPN-3-MRI and DPN-3-PET); (2) features learned by the original DPN with 6-layer networks from the original MRI and PET features, respectively (named DPN-6-MRI and DPN-6-PET); (3) features learned by SDPN from the original MRI and PET features, respectively (named SDPN-MRI and SDPN-PET); (4) features learned by SDPN from the concatenated MRI and PET features (named SDPN-MRI-PET); (5) features learned by MM-SDPN from both the original MRI and PET features.

We also compared the proposed MM-SDPN algorithm with nine state-of-the-art multimodality learning based algorithms used for AD classification as shown in Table I. It is worth noting that both the MRI and PET subsets of ADNI and the extracted ROI features in our work are the same as those data and features used in the reference [14], [18], [39], [40] and [41].

In this work, the SDPN is stacked by 2-level basic DPNs, and each basic DPN consists of 3-layer networks. Notably, the abovementioned SDPN is also used in MM-SDPN. The original DPN algorithm mainly has two parameters, namely the number of network layers and the number of hidden nodes in each layer. After set the 3-layer and 6-layer DPNs, we got the number hidden nodes in the DPN-3 algorithm by greedy search on training dataset, and then make them as the initial parameters for all the other DPN-based algorithms for further fine-tuning.

In this work, the SDPN is stacked by 2-level basic DPNs, and each basic DPN consists of 3-layer networks. Notably, the abovementioned SDPN is also used in MM-SDPN. The original DPN algorithm mainly has two parameters, namely the number of network layers and the number of hidden nodes in each layer. After set the 3-layer and 6-layer DPNs, we got the number hidden nodes in the DPN-3 algorithm by greedy search on training dataset, and then make them as the initial parameters for all the other DPN-based algorithms for further fine-tuning.

Two classifiers, namely the embedded linear classifier (LC) and the linear support vector machine (SVM) [53], are adopted

TABLE I
COMPARISON OF PERFORMANCE OF DIFFERENT MULTIMODAL CLASSIFICATION ALGORITHMS

Algorithms	Subjects	Modalities	Algorithm Description
MKL [14] *	51 AD, 43 MCI-C, 56MCI-NC, 52NC	MRI + PET + CSF	The classical MKL based algorithm
MTL [18] *	51 AD, 43 MCI-C, 56 MCI-NC, 52NC	MRI + PET + CSF	The multi-tasking learning (MTL) based algorithm
M-RBM [20]	93 AD, 76 MCI-C, 128 MCI-NC, 101 NC	MRI + PET	The pioneering multimodal deep RBM (M-RBM) based feature learning algorithms
SAE [38]	85AD,67 MCI-C,102 MCI-NC, 77 NC	MRI + PET	The SAE-based multimodal neuroimaging feature learning algorithm
SAE-MKL [39] *	51 AD, 43 MCI-C, 56 MCI-NC, 52 NC	MRI + PET + CSF	The combination of SAE-based feature learning and MKL classification (SAE-MKL) algorithm
DW-S^{2}MTL [40] *	51 AD, 43 MCI-C, 56 MCI-NC, 52 NC	MRI + PET	The deep sparse multi-task learning based feature selection (DW-S2MTL) algorithm
Dropout-DL [41] *	51 AD, 43 MCI-C,56 MCI-NC, 52 NC	MRI + PET + CSF	The dropout based robust multitask deep learning (Dropout-DL) algorithm
SDSAE [42]	94 AD, 121 MCI, 123 NC	Longitudinal MRI	The SDSAE-based feature learning algorithm
NGF [52]	37 AD, 75 MCI, 35 NC	MRI + PET + CSF + Genetics	The nonlinear graph fusion (NGF) based algorithm

The superscript * indicates that these compared algorithms use the same ADNI MRI and PET subsets.

TABLE II
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH SVM CLASSIFIER FOR AD VS. NC (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	94.02 ± 7.00	93.68 ± 10.21	94.67 ± 9.57
DPN-6-MRI	95.15 ± 6.31	94.48 ± 9.88	96.10 ± 7.95
SDPN-MRI	95.44 ± 6.43	94.33 ± 9.72	96.33 ± 8.17
DPN-3-PET	93.15 ± 6.67	92.46 ± 9.43	94.02 ± 10.97
DPN-6-PET	94.35 ± 7.08	93.73 ± 9.54	94.25 ± 10.42
SDPN-PET	95.11 ± 6.06	94.71 ± 8.62	95.70 ± 8.25
SDPN-MRI-PET	95.55 ± 6.23	95.26 ± 8.84	96.13 ± 8.82
MM-SDPN	**97.13 ± 4.44**	**95.93 ± 7.84**	**98.53 ± 5.05**

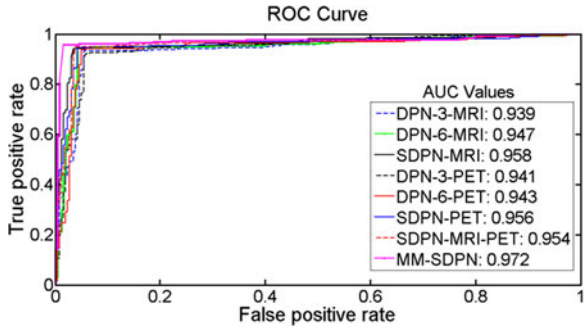

Fig. 5. ROC curves of different algorithms with the corresponding AUC values for AD vs. NC.

to more comprehensively evaluate the performance of the DPN-based features. The SVM is performed with the LIBSVM toolbox [54]. The original ROI features extracted from the single-modal MRI or PET are only processed by SVM classifier.

The 10-fold cross-validation strategy is performed for all algorithms, and this process is repeated 5 times independently so as to avoid the sampling bias introduced by randomly partitioning dataset in the cross-validation. The classification accuracy (ACC), sensitivity (SEN) and specificity (SPE) are selected as evaluation indices. The classification result is given by the format of mean ± SD (standard deviation) over all the 50 results. Moreover, the receiver operating characteristic (ROC) curve and the value of the area under ROC curve (AUC) are also used for SVM classifier.

C. Results on AD vs. NC

In the classification of AD and NC, Table II shows the classification results of different feature learning algorithms with SVM classifier. It can be found that MM-SDPN algorithm achieves the best performance with mean classification accuracy of 97.13 ± 4.44%, sensitivity of 95.93 ± 7.84% and specificity of 98.53 ± 5.05%, because it successfully fuses MRI and PET information. On the other hand, although DPN-6 (DPN with 6-layer networks) outperforms DPN-3 on both MRI and PET data for single modality imaging based AD classification, SDPN is still a little better than DPN-6, which indicates the effectiveness of SDPN due to the stacking technique.

Fig. 5 shows the ROC curves of different algorithms with their corresponding AUC values. The AUC value of MM-SDPN (0.972) is superior to others.

As shown in Table III, when applying linear classifier, different algorithms have similar tendency to those in Table II, which indicates that it is not the classifier but the SPDN plays the key role in final classification performance. The proposed MM-SDPN algorithm again obtains the best classification accuracy of 96.93 ± 4.53%, sensitivity of 95.02 ± 8.56%, and specificity of 98.37 ± 5.66%. SDPN algorithm is also superior to the original DPN. Here, DPN-6 is still superior to DPN-3 for both MRI and PET data, but SDPN achieves much better results than DPN-6.

Table IV gives the compared results of the proposed MM-SDPN and other state-of-the-art multi-modality learning based

TABLE III
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH LINEAR CLASSIFIER FOR AD VS. NC (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	90.76 ± 8.62	87.7 ± 13.99	91.93 ± 11.04
DPN-6-MRI	92.52 ± 8.14	91.56 ± 11.85	93.40 ± 9.99
SDPN-MRI	93.44 ± 7.25	90.76 ± 11.23	95.53 ± 9.63
DPN-3-PET	90.25 ± 8.92	89.24 ± 10.98	91.23 ± 12.49
DPN-6-PET	91.58 ± 7.67	89.49 ± 11.52	92.16 ± 11.46
SDPN-PET	94.78 ± 6.79	93.35 ± 9.45	96.43 ± 7.75
SDPN-MRI-PET	95.89 ± 4.88	94.46 ± 8.64	97.67 ± 6.40
MM-SDPN	**96.93 ± 4.53**	**95.02 ± 8.56**	**98.37 ± 5.66**

TABLE IV
COMPARISON OF MM-SDPN WITH MULTI-MODALITY FEATURE LEARNING BASED ALGORITHMS FOR AD VS. NC (UNIT: %)

	ACC	SEN	SPE
MKL [14] *	93.20	93.00	93.30
MTL [18] *	95.38	94.71	95.77
M-RBM [20]	95.35 ± 5.23	94.65	95.22
SAE [38]	91.40 ± 5.56	92.32 ± 6.29	90.42 ± 6.93
SAE-MKL [39] *	95.90 ± 1.10	/	/
DW-S^2MTL [40] *	93.18	90.00	96.33
Dropout-DL [41] *	91.40	/	/
SDSAE [42]	91.95 ± 1.00	89.49 ± 2.37	93.82 ± 1.63
NGF [52]	91.80	/	/
MM-SDPN-SVM	**97.13 ± 4.44**	**95.93 ± 7.84**	**98.53 ± 5.05**
MM-SDPN-LC	96.93 ± 4.53	95.02 ± 8.56	98.37 ± 5.66

The superscript * indicates that these compared algorithms use the same ADNI MRI and PET subsets.0

algorithms. MM-SDPN outperforms all other algorithms with both SVM and linear classifier on all evaluation indices. MM-SDPN with SVM improves at least 1.23%, 1.22% and 2.20% on accuracy, sensitivity and specificity, respectively, compared with other non-SDPN multi-modality learning algorithms, which indicates the effectiveness of proposed MM-SDPN algorithm to fuse and learn feature representation from MRI and PET data for AD classification. It is also worth noting that the proposed MM-SDPN algorithm is superior to these algorithms in [14], [18], [39], [40] and [41] on the same MRI and PET subsets of ADNI with the same ROI features.

D. Results on MCI vs. NC

Tables V and VI show the results of different DPN-based algorithms for the classification of MCI vs. NC with SVM and linear classifier, respectively. As can be seen, MM-SDPN still outperforms all other methods with both SVM and linear classifier. The best classification accuracy, sensitivity and specificity are 87.24 ± 4.52%, 97.91 ± 4.17% and 67.04 ± 9.29%, respectively, with SVM, and 86.99 ± 4.82%, 94.24 ± 6.16% and 71.32 ± 9.93%, respectively, with linear classifier. SDPN is superior to DPN for single modality imaging based MCI classification on both MRI and PET data, although DPN-6 outperforms DPN-3.

Fig. 6 shows the ROC curves of different algorithms with their corresponding AUC values, and MM-SDPN achieves the second performance with ROC value of 0.901.

TABLE V
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH SVM CLASSIFIER FOR MCI VS. NC (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	80.92 ± 6.36	90.47 ± 8.32	62.60 ± 11.35
DPN-6-MRI	82.11 ± 6.54	91.07 ± 7.35	64.87 ± 10.99
SDPN-MRI	83.29 ± 5.85	92.87 ± 7.34	64.93 ± 12.98
DPN-3-PET	83.78 ± 5.26	94.51 ± 6.53	63.20 ± 13.20
DPN-6-PET	84.10 ± 5.37	93.96 ± 6.42	65.33 ± 13.50
SDPN-PET	84.64 ± 4.05	95.13 ± 5.87	64.20 ± 9.50
SDPN-MRI-PET	85.81 ± 4.69	95.73 ± 5.82	66.93 ± 9.25
MM-SDPN	**87.24 ± 4.52**	**97.91 ± 4.17**	**67.04 ± 9.29**

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH LINEAR CLASSIFIER FOR MCI VS. NC (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	80.81 ± 7.32	84.47 ± 8.11	68.40 ± 10.22
DPN-6-MRI	81.85 ± 7.35	85.64 ± 10.64	68.60 ± 9.69
SDPN-MRI	83.16 ± 4.44	89.69 ± 6.55	69.07 ± 10.39
DPN-3-PET	81.42 ± 7.73	90.02 ± 9.03	65.13 ± 11.29
DPN-6-PET	83.17 ± 5.40	90.69 ± 7.33	67.67 ± 9.90
SDPN-PET	84.51 ± 4.64	91.53 ± 7.94	68.00 ± 9.31
SDPN-MRI-PET	85.31 ± 4.81	92.56 ± 6.35	70.93 ± 9.97
MM-SDPN	**86.99 ± 4.82**	**94.24 ± 6.16**	**71.32 ± 9.93**

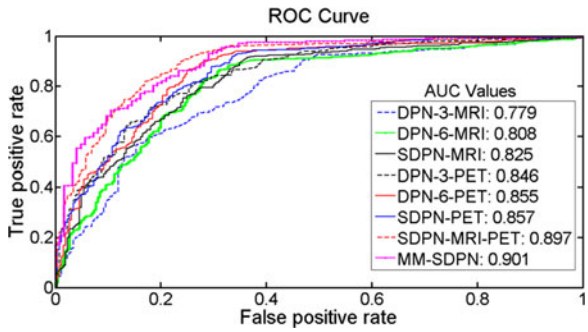

Fig. 6. ROC curves of different algorithms with the corresponding AUC values for MCI vs. NC.

It is worth noting that in both Tables V and VI, the sensitivity is much higher than the specificity for MCI vs. NC due to the sample unbalance problem, where the samples of MCI patients are double of NC subjects. On the other hand, the results of PET are better than those of MRI for MCI vs. NC, which may be explained by the theoretical models of temporal ordering of relative biomarkers [55]. Specifically speaking, the changes in FDG-PET metabolic measures precede the changes in the brain structures of MRI for MCI patient [55].

As shown in Table VII, MM-SDPN once again outperforms the compared multi-modality learning based algorithms. MM-SDPN with SVM makes more than 1.57% and 2.54% improvements on accuracy and sensitivity, respectively compared with the other non-SDPN algorithms. Only the specificity of SAE in [38] is very high, but with very low accuracy and sensitivity.

TABLE VII
COMPARISON OF MM-SDPN WITH MULTI-MODALITY FEATURE LEARNING BASED ALGORITHMS FOR MCI VS. NC (UNIT: %)

	ACC	SEN	SPE
MKL [14] *	76.40	81.80	66.00
MTL [18] *	82.99	89.39	70.77
M-RBM [20]	85.67 ± 5.22	95.37	65.87
SAE [38]	82.10 ± 4.91	60.00 ± 13.93	**92.32 ± 8.74**
SAE-MKL [39] *	85.00 ± 1.20	/	/
DW-S^{2}MTL [40] *	80.11	93.89	53.67
Dropout-DL [41] *	77.40	/	/
SDSAE [42]	83.72 ± 1.16	84.74 ± 2.34	82.72 ± 1.19
NGF [52]	79.50	/	/
MM-SDPN-SVM	**87.24 ± 4.52**	**97.91 ± 4.17**	67.04 ± 9.29
MM-SDPN-LC	86.99 ± 4.82	94.24 ± 6.16	71.32 ± 9.93

The superscript * indicates that these compared algorithms use the same ADNI MRI and PET subsets.

TABLE VIII
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH SVM CLASSIFIER FOR MCI-C VS. MCI-NC (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	71.85 ± 7.91	58.60 ± 10.40	82.27 ± 11.22
DPN-6-MRI	73.82 ± 6.59	60.10 ± 10.86	84.27 ± 9.92
SDPN-MRI	75.47 ± 5.56	63.00 ± 10.64	85.07 ± 7.54
DPN-3-PET	71.18 ± 5.59	59.60 ± 10.73	80.13 ± 8.83
DPN-6-PET	73.34 ± 5.69	60.90 ± 10.82	82.93 ± 8.46
SDPN-PET	74.43 ± 5.73	62.10 ± 11.12	83.80 ± 7.74
SDPN-MRI-PET	76.61 ± 5.79	64.60 ± 11.15	85.87 ± 7.27
MM-SDPN	**78.88 ± 4.38**	**68.04 ± 9.99**	**86.81 ± 9.12**

TABLE IX
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH LINEAR CLASSIFIER FOR MCI-C VS. MCI-NC (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	68.34 ± 6.65	57.12 ± 10.00	76.81 ± 9.30
DPN-6-MRI	69.67 ± 6.37	57.7 ± 10.51	77.20 ± 9.14
SDPN-MRI	71.80 ± 5.45	57.6 ± 10.75	80.73 ± 7.69
DPN-3-PET	68.94 ± 8.04	58.6 ± 11.56	76.93 ± 9.59
DPN-6-PET	70.68 ± 6.08	59.00 ± 10.05	79.67 ± 8.34
SDPN-PET	72.82 ± 5.62	61.00 ± 10.55	81.80 ± 8.31
SDPN-MRI-PET	74.33 ± 5.67	61.50 ± 10.61	84.07 ± 7.36
MM-SDPN	**76.52 ± 5.99**	**62.50 ± 10.65**	**86.27 ± 7.49**

E. Results on MCI-C vs. MCI-NC

It is known that the classification of MCI-C from MCI-NC is usually difficult as shown in previous work [18], [20], [41]. Tables VIII and IX give the results of different DPN-based algorithms on MCI-C vs. MCI-NC with different classifiers. It can be observed that the best results are again achieved by MM-SDPN, whose mean classification accuracy, sensitivity and specificity are 78.88 ± 4.38%, 68.04 ± 9.99% and 86.81 ± 9.12%, respectively, with SVM classifier, and 76.52 ± 5.99%, 62.50 ± 10.65% and 86.27 ± 7.49% respectively, with linear classifier. SDPN gets much better performance than both DPN-6 and DPN-3 for single modality neuroimaging data for this difficult task.

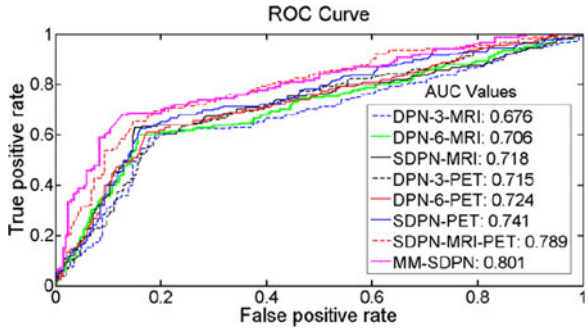

Fig. 7. ROC curves of different algorithms with the corresponding AUC values for MCI-C vs. MCI-NC.

TABLE X
COMPARISON OF MM-SDPN WITH MULTI-MODALITY FEATURE LEARNING BASED ALGORITHMS FOR MCI-C VS. MCI-NC (UNIT: %)

	ACC	SEN	SPE
MTL [18] *	72.28	66.05	76.61
M-RBM [20]	75.92 ± 15.37	48.04	**95.23**
SAE-MKL [39] *	75.80 ± 2.00	/	/
DW-S^{2}MTL [40] *	74.15	50.50	92.67
Dropout-DL [41]*	57.40	/	/
MM-SDPN-SVM	**78.88 ± 4.38**	**68.04 ± 9.99**	86.81 ± 9.12
MM-SDPN-LC	76.52 ± 5.99	62.50 ± 10.65	86.27 ± 7.49

The superscript * indicates that these compared algorithms use the same ADNI MRI and PET subsets.

TABLE XI
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH SVM CLASSIFIER FOR MULTICLASS CLASSIFICATION (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	50.21 ± 4.80	48.61 ± 4.90	83.17 ± 1.63
DPN-6-MRI	51.17 ± 5.52	49.80 ± 5.59	83.55 ± 2.01
SDPN-MRI	52.86 ± 5.08	50.23 ± 5.58	83.70 ± 1.85
DPN-3-PET	48.50 ± 4.33	46.99 ± 4.26	82.58 ± 1.43
DPN-6-PET	49.69 ± 4.95	48.11 ± 4.81	83.00 ± 1.63
SDPN-PET	51.47 ± 4.85	49.58 ± 4.56	83.59 ± 1.49
SDPN-MRI-PET	53.79 ± 4.42	51.54 ± 4.81	84.21 ± 1.62
MM-SDPN	**57.00 ± 3.65**	**53.65 ± 4.04**	**85.05 ± 1.39**

In Fig. 7 shows the ROC curves of different algorithms with their corresponding AUC values, and MM-SDPN again achieves the best performance with AUC value of 0.801.

As can be seen from Table X, for such a difficult classification task, the proposed MM-SDPN still outperforms all other non-SDPN multi-modality feature learning based algorithms, improving the accuracy and sensitivity at least 2.96% and 1.99%, respectively, with SVM classifier. Deep RBM based algorithm achieves the best specificity, but lower sensitivity.

F. Results on AD vs. MCI-C vs. MCI-NC vs. NC

Multiclass classification of AD vs. MCI-C vs. MCI-NC vs. NC is an extremely tough task. The results in Tables XI and XII give similar trend to other binary classification, showing that MM-SDPN performs the best with classification accuracy

TABLE XII
CLASSIFICATION RESULTS OF DIFFERENT FEATURES WITH LINEAR CLASSIFIER FOR MULTICLASS CLASSIFICATION (UNIT: %)

	ACC	SEN	SPE
DPN-3-MRI	49.74 ± 5.83	48.42 ± 6.81	82.90 ± 2.42
DPN-6-MRI	51.54 ± 6.07	50.96 ± 4.93	83.74 ± 1.73
SDPN-MRI	52.13 ± 5.35	49.08 ± 9.14	83.10 ± 3.02
DPN-3-PET	48.42 ± 5.73	47.17 ± 3.48	82.51 ± 1.34
DPN-6-PET	49.97 ± 5.23	48.25 ± 2.21	82.93 ± 0.92
SDPN-PET	51.06 ± 6.33	49.23 ± 6.91	83.12 ± 2.30
SDPN-MRI-PET	53.36 ± 5.06	50.58 ± 7.83	83.77 ± 2.35
MM-SDPN	**55.34 ± 4.57**	**52.49 ± 6.12**	**84.18 ± 1.99**

TABLE XIII
COMPARISON OF MM-SDPN WITH MULTI-MODALITY FEATURE LEARNING BASED ALGORITHMS FOR MULTICLASS CLASSIFICATION (UNIT: %)

	ACC	SEN	SPE
SAE [38]	53.79 ± 4.76	52.14 ± 11.81	**86.98 ± 9.62**
MM-SDPN-SVM	**57.00 ± 3.65**	**53.65 ± 4.04**	85.05 ± 1.39
MM-SDPN-LC	55.34 ± 4.57	52.49 ± 6.12	84.18 ± 1.99

of 57.00 ± 3.65%, sensitivity of 53.65 ± 4.04% and specificity of 85.05 ± 1.39% by SVM, and the corresponding results of 55.34 ± 4.57%, 52.49 ± 6.12% and 84.18 ± 1.99% with linear classifier. Again, SDPN is still superior to DPN.

Moreover, as shown in Table XIII, compared with the state-of-the-art SAE-based algorithm [38], our proposed MM-SDPN algorithm with SVM classifier achieves the improvement of 3.21% on classification accuracy and 1.51% on sensitivity.

IV. DISCUSSION

In this work, we propose a MM-SDPN algorithm to effectively learn features for multimodal neuroimaging based AD diagnosis. The results on the ADNI dataset with four groups of experiments show that the proposed MM-SDPN algorithm achieves the best performance in comparison with the state-of-the-art multi-modality learning based algorithms. Specifically,

In our study, the original ROI features are low-level features, which cannot represent the attributes of AD in a well-differentiated way. When applying DPN to learn features from ROI features, more complex representation is provided, and therefore significant improvements have been achieved by DPN. After stacking the basic DPN block several times, higher level representation is obtained. Therefore, SDPN achieves better performance than the original DPN for single-modal neuroimaging based AD classification. Notably, although the 6-layer DPN is superior to the 3-layer DPN with the increase of hidden layers, too deep network will increase the computational complexity while the accuracy of approximation has not obvious increment according to the universal approximation theorem in neural network. On the other hand, the results also show that SDPN with two 3-layer DPNs outperforms the 6-layer DPN, because the basis of the first layer in second-level basic DPN is built on higher-level features, namely the concatenation features of the first-level basic DPN, and this basis will generate more effective and higher-level features after the learning of the second-level DPN. Moreover, SDPN is easier to tune parameters than DPN with deeper networks to achieve same performance.

Since fusion of multimodal neuroimaging data can effectively benefit the classification performance for AD diagnosis, a MM-SDPN algorithm is then proposed, which is composed of two-stage SDPN. Two SDPNs are applied to the ROI features of MRI and PET, respectively, to get abstract of each modality in the first stage. Both learned features are then concentrated and fed to a new SDPN to learn the fused features, which include both the intrinsic properties of each modality and the correlations among MRI and PET. Compared with the way of directly applying single SDPN to the concentrated ROI features of MRI and PET, the high-level features learned by the first stage DPNs will benefit and improve the learning performance of the second stage SDPN in MM-SDPN, and therefore, MM-SDPN has more effective ability to learn and fuse multimodal neuroimaging data.

In this study, two classifiers, namely SVM and linear classifier, are used to evaluate the performance of SDPN and MM-SDPN. Both classifiers give similar results, which indicate that the well performance for AD classification more depends on the learned features instead of classifier. Therefore, MM-SDPN truly effectively learns a good feature representation.

Three main properties of DPN contribute to the excellent performance of MM-SDPN as follows. (1) The neuroimaging dataset usually has small labeled samples. When applying DPN to such a small dataset, its constructed networks have small nodes, which guarantee a well-trained deep network with small samples [42]. (2) In DPN networks, the nodes of the first k layers form a basis of all values attained by degree-k polynomials. Therefore, the DPN network might have a large bias but will tend not to overfit (i.e. low variance), and even for deeper network, the bias is gradually decreased while the variance is increased. Thus, in principle, the overfitting can be depressed by controlling the bias-variance tradeoff [43]. Moreover, the intermediate layer's connections in DPN are very sparse, i.e. each node in the intermediate layers is limited to connect to only few other nodes, rather than all nodes in the previous layer, which can prevent the overfitting [43]. Consequently, the algorithm structure of DPN makes it appropriate for small dataset. (3) Since the neuroimaging data usually provide limited labeled ground truth samples only, and the prior label information is beneficial for classification task with small data, the supervised DPN is more suitable for small neuroimaging dataset than the unsupervised DL algorithms. Moreover, the proposed MM-SDPN outperforms the fine-turning-based supervised SAE algorithms in [38] and [39] as shown in the work.

DPN is a new DL algorithm, and the theory and algorithm improvement about it is still scarce. Therefore, in future work, we will not only further improve DPN algorithm, but also pay more attention to analyze the framework of DPN, especially the difference between DPN and other DL algorithms. On the other hand, the proposed MM-SDPN algorithm in this work shows its effectiveness for small dataset. In fact, DPN has already gained its good reputation for learning features from large-scale data in [43]. Since there are no forward and backward feedbacks

between successive basic DPN, SDPN and MM-SDPN are relatively simple and fast. Therefore, they are also expected to be effective for large-scale data. In a future work, we plan to apply MM-SDPN to learn feature representation directly from local patches of MRI and PET, which is similar to the method in [20]. Moreover, semi-supervised MM-SDPN will also be studied in future, since it is relatively easy to acquire the unlabeled medical images that are helpful to improve the performance of representation learning. Motivated by the successful application of MKL to multimodal neuroimaging data, we will try to combine MKL with MM-SDPN for AD classification. The learned features from the second stage SDPN and the individual learned features from MRI and PET in the first stage SDPNs can be regarded as multi-view data for MKL. Thus, CSF features can also be embedded to MKL, which will potentially improve the performance for AD classification with the MM-SDPN and MKL based framework.

V. Conclusion

In this work, a MM-SDPN algorithm has been proposed. It consists of two-stage SDPN, and can effectively learn and fuse multimodal data for diagnosis of Alzheimer's disease. MM-SDPN achieves the state-of-the-art performance for classifying both two stages and four stages of AD progression. Therefore, the proposed MM-SDPN can be a powerful representation algorithm for not only multimodal neuroimaging data but also other medical data.

Acknowledgment

The authors would like to thank the Alzheimer's Disease Neuroimaging Initiative for providing data for this paper.

Reference

[1] G. McKhann, D. Drachman, M. Folstein, R. Katzman, D. Price, and E. M. Stadlan, "Clinical diagnosis of Alzheimer's disease report of the NINCDSADRDA work group under the auspices of department of health and human services task force on Alzheimer's disease," *Neurology*, vol. 34, no. 7, pp. 939–939, 1984.

[2] R. Brookmeyer, E. Johnson, K. Ziegler-Graham, and H. Michael Arrighi, "Forecasting the global burden of Alzheimer's disease," *Alzheimers Dementia*, vol. 3, no. 3, pp. 186–191, 2007.

[3] J. A. Turner, S. G. Potkin, G. G. Brown, D. B. Keator, G. McCarthy, and G. H. Glover, "Neuroimaging for the diagnosis and study of psychiatric disorders," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 112–117, Jul. 2007.

[4] S. D. Liu *et al.*, "Multimodal neuroimaging computing: A review of the applications in neuropsychiatric disorders," *Brain Informat.*, vol. 2, pp. 167–180, 2015.

[5] L. Alves, A. S. A. Correia, R. Miguel, P. Alegria, and P. Bugalho, "Alzheimer's disease: A clinical practice-oriented review," *Front. Neurol.*, vol. 3, pp. 1–20, 2012.

[6] S. L. Risacher and A. J. Saykin, "Neuroimaging and other biomarkers for Alzheimer's disease: The changing landscape of early detection," *Annu. Rev. Clin. Psychol.*, vol. 9, pp. 621–648, 2013.

[7] R. M. Ahmed*et al.*, "Biomarkers in dementia: Clinical utility and new directions," *J. Neurol. Neurosurg. Psych.*, vol. 85, pp. 1426–1434, 2014.

[8] R. Cuingnet*et al.*, "Automatic classification of patients with Alzheimer's disease from structural MRI: A comparison of ten methods using the ADNI database," *NeuroImage*, vol. 56, no. 2, pp. 766–781, 2011.

[9] F. Falahati, E. Westman, and A. Simmons, "Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging," *J. Alzheimers Dis.*, vol. 41, no. 3, pp. 685–708, 2014.

[10] B. Mwangi, T. S. Tian, and J. C. Soares, "A review of feature reduction techniques in neuroimaging," *Neuroinformatics*, vol. 12, pp. 229–244, 2014.

[11] K. Uludag, and A. Roebroeck, "General overview on the merits of multimodal neuroimaging data fusion," *NeuroImage*, vol. 102, pp. 3–10, 2014.

[12] B. K. Cao, X. N. Kong, and P. S. Yu, "A review of heterogeneous data mining for brain disorder identification," *Brain Informat.*, vol. 2, no. 4, pp. 253–264, 2015.

[13] C. Hinrichs, V. Singh, G. F. Xu, S. C. Johnson, and ADNI, "Predictive markers for AD in a multi-modality framework: An analysis of MCI progression in the ADNI population," *NeuroImage*, vol. 55, no. 2, pp. 574–589, 2011.

[14] D. Q. Zhang, Y. P. Wang, L. P. Zhou, H. Yuan, D. G. Shen, and ADNI, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *NeuroImage*, vol. 55, pp. 856–867, 2011.

[15] K. R. Gray, P. Aljabar, R. A. Heckemann, A. Hammers, D. Rueckert, and ADNI, "Random forest-based similarity measures for multi-modal classification of Alzheimer's disease," *NeuroImage*, vol. 65, pp. 167–175, 2013.

[16] F. Y. Liu, L. P. Zhou, C. H. Shen, and J. P. Yin, "Multiple kernel learning in the primal for multi-modal Alzheimer's disease classification," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 3, pp. 984–990, May 2014.

[17] M. Dyrba, M. Grothe, T. Kirste, and S. J. Teipel, "Multimodal analysis of functional and structural disconnection in Alzheimer's disease using multiple kernel SVM," *Hum. Brain Mapping*, vol. 36, no. 6, pp. 2118–2131, 2015.

[18] B. Jie, D. Q. Zhang, B. Cheng, D. G. Shen, and ADNI, "Manifold regularized multitask feature learning for multimodality disease classification," *Hum. Brain Mapping*, vol. 36, pp. 489–507, 2015.

[19] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[20] H.I. Suk, S. W. Lee, D. G. Shen, and ADNI, "Hierarchical feature representation and multimodal fusion with deep learning for AD MCI diagnosis," *NeuroImage*, vol. 101, pp. 569–582, Nov. 2014.

[21] M. H. Liu, D. Q. Zhang, and D.G.Shen, "Ensemble sparse classification of Alzheimer's disease," *NeuroImage*, vol. 60, pp. 1106–1116, 2012.

[22] P. Coupé *et al.*, "Scoring by nonlocal image patch estimator for early detection of Alzheimer's disease," *NeuroImage, Clin.*, vol. 1, pp. 141–152, 2012.

[23] A. Gupta, M. Ayhan, and A. Maida, "Natural image bases to represent neuroimaging data," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 987–994.

[24] A. Payan and G. Montana, "Predicting Alzheimer's disease: A neuroimaging study with 3D convolutional neural networks," in *Proc. Int. Conf. Pattern Recog. Appl. Methods*, 2015.

[25] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[26] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[27] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 145–154, Jan. 2011.

[28] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[29] G. Carneiro, J. C. Nascimento, and A. Freitas, "The segmentation of the left ventricle of the heart from ultrasound data using deep learning architectures and derivative-based search methods," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 968–982, Mar. 2012.

[30] D. C. Cires, L. M. Gambardella, A. Giusti, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. Neural Inf. Process. Syst. Conf.*, 2012, pp. 2852–2860.

[31] H. C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[32] G. Carneiro and J. C. Nascimento, "Combining multiple dynamic models and deep learning architectures for tracking the left ventricle endocardium in ultrasound data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2592–2607, Nov. 2013.

[33] H. Roth *et al.*, "A new 2.5D representation for lymph node detection using random sets of deep convolutional," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2014, pp. 520–527.

[34] M. P. Sergey *et al.*, "Deep learning for neuroimaging: A validation study," *Front. Neurosci.*, vol. 8, pp. 1–11, 2014.

[35] W. L. Zhang *et al.*, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214–224, 2015.
[36] T. Brosch, R. Tam, and ADNI, "Manifold learning of brain MRIs by deep learning," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2013, pp. 633–640,.
[37] V. K. Ithapu, V. Singh, O. Okonkwo, and S. C. Johnson, "Randomized denoising autoencoders for smaller and efficient imaging based AD clinical trials," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2014, pp. 470–478,.
[38] S. Q. Liu *et al.*, "Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 4, pp. 1132–1140, Apr. 2015.
[39] H. Suk, and D. Shen, "Deep learning-based feature representation for AD/MCI classification," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2013, pp. 583–590.
[40] H. I. Suk, S. W. Lee, D. Shen, and ADNI, "Deep sparse multi-task learning for feature selection in Alzheimer's disease diagnosis," *Brain Struct. Funct.*, vol. 221, pp. 1–19, 2015.
[41] F. Li, L. Tran, K. H. Thung, S. W. Ji, D. G. Shen, and J. Li, "A robust deep model for improved classification of AD/MCI patients," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 5, pp. 1610–1616, Sep. 2015.
[42] B. Shi, Y. Chen, P. Zhang, C. D. Smith, and J. Liu, "Nonlinear feature transformation and deep fusion for Alzheimer's disease staging analysis," *Pattern Recog.*, vol. 63, pp. 487–498, 2017.
[43] R. Livni, S. Shalev-Shwartz, and O. Shamir, "An algorithm for training polynomial networks," to be published.
[44] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proc. Neural Inf. Process. Syst. Conf.*, 2006, pp. 153–160.
[45] C. R. Jack *et al.*, "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods," *J. Magn. Reson. Imag.*, vol. 27, pp. 685–691, 2008.
[46] J. G. Sled, A. P. Zijdenbos, and A. C. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in MRI data," *IEEE Trans. Med. Imag.*, vol. 17, no. 1, pp. 87–97, Feb. 1998.
[47] Y. P. Wang, J. X. Nie, P. T. Yap, F. Shi, L. Guo, and D. G. Shen, "Robust deformable-surface-based skull-stripping for large-scale studies," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2011, pp. 635–642.
[48] Y. P. Wang *et al.*, "Knowledge-guided robust MRI brain extraction for diverse large-scale neuroimaging studies on humans and non-human primates," *Plos One*, vol. 9, p. e77810, 2014.
[49] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, Jan. 2001.
[50] D. G. Shen and C. Davatzikos, "HAMMER: Hierarchical attribute matching mechanism for elastic registration," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1421–1439, Nov. 2002.
[51] N. Kabani, D. MacDonald, C. J. Holmes, and A. Evans, "A 3D atlas of the human brain," *NeuroImage*, vol. 7, p. S717, 1998.
[52] T. Tong, K. Gray, Q Gao, L. Chen, D. Rueckert, and ADNI, "Multi-modal classification of Alzheimer's disease using nonlinear graph fusion," *Pattern Recog.*, vol. 63, pp. 171–181, 2017.
[53] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.
[54] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1—39, 2001.
[55] C. R. Jack *et al.*, "Hypothetical model of dynamic biomarkers of the Alzheimer's pathological cascade," *Lancet Neurol.*, vol. 9, pp. 119–128, 2010.