HHS Public Access

Author manuscript

Med Image Comput Comput Assist Interv. Author manuscript; available in PMC 2015 June 13.

Published in final edited form as:

Med Image Comput Comput Assist Interv. 2014 ; 17(0 3): 305–312.

Deep Learning Based Imaging Data Completion for Improved Brain Disease Diagnosis

Rongjian Li[1], **Wenlu Zhang**[1], **Heung-Il Suk**[2], **Li Wang**[2], **Jiang Li**[3], **Dinggang Shen**[2], and **Shuiwang Ji**[1]

[1]Department of Computer Science, Old Dominion University, Norfolk, VA, USA

[2]Department of Radiology, University of North Carolina at Chapel Hill, NC, USA

[3]Department of ECE, Old Dominion University, Norfolk, VA, USA

Abstract

Combining multi-modality brain data for disease diagnosis commonly leads to improved performance. A challenge in using multi-modality data is that the data are commonly incomplete; namely, some modality might be missing for some subjects. In this work, we proposed a deep learning based framework for estimating multi-modality imaging data. Our method takes the form of convolutional neural networks, where the input and output are two volumetric modalities. The network contains a large number of trainable parameters that capture the relationship between input and output modalities. When trained on subjects with all modalities, the network can estimate the output modality given the input modality. We evaluated our method on the Alzheimer's Disease Neuroimaging Initiative (ADNI) database, where the input and output modalities are MRI and PET images, respectively. Results showed that our method significantly outperformed prior methods.

1 Introduction

Alzheimer's disease (AD) is a common neuro-degenerative disease for which we still lack effective treatment. It has been shown that early detection and intervention at its prodromal stage, such as the mild cognitive impairment (MCI) stage, are effective in delaying the onset of AD. Developments in neuroimaging techniques, such as the magnetic resonance imaging (MRI) and positron emission tomography (PET) techniques, coupled with advanced computational methods, have led to accurate prediction of AD and MCI [1].

A key challenge in employing computational methods for disease diagnosis is that the neuroimaging data usually consist of multiple modalities, but they could be incomplete in the sense that not all subjects have all data modalities. The accuracy of disease diagnosis might be improved if the missing data could be estimated. However, the relationship between different data modalities is complicated and nonlinear. Thus, a highly sophisticated model is required for the collaborative completion of neuroimaging data.

Deep convolutional neural networks (CNNs) are a type of multi-layer, fully trainable models that are capable of capturing highly nonlinear mappings between inputs and outputs [2]. These models were originally motivated from computer vision problems and thus are intrinsically suitable for image-related applications. Deep CNNs have been successfully applied to a variety of applications, including image classification [2,3], segmentation [4], and denoising [5].

In this work, we propose to use deep CNNs for completing and integrating multi-modality neuroimaging data. Specifically, we designed a 3-dimensional (3-D) CNN architecture that takes one volumetric data modality as input and another volumetric data modality as its output. When trained end-to-end on subjects with both data modalities, the network captures the nonlinear relationship between two data modalities. This allows us to predict and estimate the output data modality given the input modality.

We applied our 3-D CNN model to predict the missing PET patterns from the MRI data. We trained our model on subjects with both PET and MRI data, where the MRI data were used as input and the PET data were used as output. The trained network contains a large number of parameters that encode the nonlinear relationship between MRI and PET data. We used the trained network to estimate the PET patterns for subjects with only MRI data. Results showed that our method outperformed prior methods on disease diagnosis.

2 Material and Methods

2.1 Data Preprocessing

The data used in this work were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. For each subject, the T1-weighted MRI was processed by correcting the intensity inhomogeneity followed by skull-stripping and cerebellum removing. In addition, each MRI was segmented into gray matter, white matter and cerebrospinal fluid and was further spatially normalized into a template space. In this work, the gray matter tissue density maps were used. The PET images were also obtained from ADNI, and they were rigidly aligned to the respective MR images. The gray matter tissue density maps and the PET images were further smoothed using a Gaussian kernel (with unit standard deviation) to improve the signal-to-noise ratio. To reduce the computational cost, we downsampled both the gray matter tissue density maps and PET images to $64 \times 64 \times 64$ voxels.

We used data for 830 subjects in the ADNI baseline data set. This data set was acquired from 198 AD patients, 403 MCI patients, which include 167 pMCI patients (who will progress to AD in 18 months) and 236 sMCI patients (whose symptom were stable and will not progress to AD in 18 months), and 229 healthy normal controls (NC). Out of these 830 subjects, more than half of them (432) do not have PET images. Thus, accurate completion of PET images for these subjects would improve the accuracy of disease diagnosis.

2.2 3-D Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of deep models that are able to capture highly nonlinear relationships between input and output [2]. In image classification tasks,

two types of layers, *i.e.*, convolutional layer and subsampling layer, are usually stacked alternatingly. The convolutional layer applies trainable filers to feature maps in the previous layer, while the subsampling layer is used to reduce the resolution of feature maps.

CNNs have been primarily applied to 2-D images such as visual object recognition [2,3] and segmentation [6]. In [4,5], 2-D CNNs have been extended to segment and restore 3-D images. In [7], 3-D CNNs have been applied to process spatiotemporal video data. Similar to the 2-D case, 3-D CNNs perform nonlinear mapping by computing convolutions with 3-D filters.

Formally, let the value at position (*x*, *y*, *z*) on the *j*th feature map in the *i*th layer be v_{ij}^{xyz}. Then the 3-D convolution is given by

$$v_{ij}^{xyz}=\sigma\left(b_{ij}+\sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1}w_{ijm}^{pqr}v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right), \quad (1)$$

where $\sigma(\cdot)$ is the sigmoid function, b_{ij} is the bias, *m* indexes the set of feature maps in the (*i* − 1)th layer connected to the current feature map, P_i, Q_i and R_i are the sizes of the 3-D kernel along three spatial dimensions respectively, w_{ijm}^{pqr} is the (*p*, *q*, *r*)th value of the filter connected to the *m*th feature map in the previous layer. Note that Eq. (1) describes a generic 3-D convolution operation and can be applied to any layer of a 3-D CNN architecture with any number of feature maps.

Subsampling layers are commonly used in recognition and classification tasks. In these layers, the resolution of feature maps is reduced by pooling over local neighborhood, thereby enhancing invariance to distortions on the inputs. In this work, our primary focus is data completion instead of recognition. Thus, subsampling layers were not used.

2.3 3-D CNN for Imaging Data Completion

Based on the 3-D convolution described above, a variety of CNN architectures can be devised. In the following, we describe a 3-D CNN architecture, shown in Fig. 1, for PET image completion. The data for training this CNN model consist of patches extracted from subjects having both PET and MRI images. The input patch size was determined by the size of output patch in the output layer, since each convolution operation reduces the size of feature map along each dimension by a factor related to the size of filter. In this work, the size of output patches was set to 3 × 3 × 3. We randomly selected a large number of patches from each 3-D MRI volume, and the corresponding PET image patches were also obtained. Patches that cross the boundary or are located completely within background were removed. The total number of patches extracted from each volume was 50, 000 so that the entire volume is largely covered.

In the CNN architecture, we first applied 3-D convolution with a filter size of 7 × 7 × 7 on the input patch and construct 10 feature maps in the first hidden layer. The second hidden layer is again a 3-D convolution layer with 10 feature maps fully connected to all the feature maps in the previous layer. The output layer contains only one feature map, which is the

corresponding PET image patch. In addition, the filter size for mapping the feature maps of the last hidden layer to the output was set to 1 to reduce the computational cost. In total, the number of trainable parameters for this network is 37, 761. The latent nonlinear relationship between the MRI and PET images was encoded into the large number of parameters in the network. This CNN architecture was selected based on a balance between the representation power and the computational cost of training the network. A network with more layers and feature maps might be able to represent the training data better, but the computational cost of training more complex networks is prohibitive.

In this work, the CNS package [8] was used to implement the CNN architecture. The weights of this network were updated by error back-propagation using stochastic gradient descent algorithm. The learning rate was fixed to 10^{-2} in all the experiments, and other parameters were set to the default values given in the CNS package [8]. The network was trained for multiple epochs, where each epoch involves training the network by each example once. In this paper, we trained the network for 10 epochs since the performance seems to have converged after 10 epochs and the training was very time-consuming. In particular, we have $398 \times 50,000 = 19.9$ million training patches. Each epoch took about 48 hours if all the patches were used on a Tesla K20c GPU with 2,496 cores.

3 Results and Discussion

3.1 Experimental Setup

In the experiments, we focused on evaluating our 3-D CNN model for missing PET data completion. We used several controlled experiments to compare the predicted and the true PET image data. We did not employ advanced feature extraction and classification methods to compare the completed and true data, but rather used a set of standard methods to make the comparison straightforward. We consider three binary-class classification tasks (*i.e.*, AD vs. NC, MCI vs. NC, and sMCI vs. pMCI) in this paper, where MCI includes both pMCI and sMCI.

We compared our method with two other commonly used missing data estimation methods, namely, *K*-nearest neighbor (KNN) and Zero methods [9]. The experiments in this work consist of two steps. The first step is to complete the missing PET data using CNN, KNN, or Zero methods. The second step then evaluate the classification performance based on reconstructed data using the ℓ_2-norm regularized logistic regression classifiers for all methods. In the experiments, we trained the classifiers by randomly selecting 2⁄3 of the samples and performed an evaluation using the remaining 1⁄3 as test data in the second step. To obtain robust performance estimates, we repeated the random partition 30 times and reported the statistics computed over these 30 trials. Note that no class information was used in our CNN training. Thus, we built one CNN model and applied it for all 30 random trials. We performed feature selection by removing voxels that have zero value for all subjects. Since the number of samples was not balanced between classes, we used the area under the ROC curve (AUC) as the performance measure to evaluate different methods in this study.

3.2 Evaluation on Subjects with Both MRI and PET

We first evaluated whether the predicted PET data were similar to the true PET data. In the data set used for this study, there were 398 subjects with both MRI and PET images. We randomly sampled 1/2 of these 398 subjects for training the 3-D CNN model. Then the model was used to predict the PET images of the remaining 1/2 subjects. Since we had true PET images for the test subjects, we were able to compare the true and the predicted PET images both visually and quantitatively.

We first visually examined the predicted PET patterns with the ground truth data for each subject. Figure 2 shows the predicted and the ground truth data slice by slice for two subjects. We can observe that the predicted PET patterns are similar to the ground truth. This demonstrates that our deep learning based method can successfully estimate the missing PET data.

To evaluate the proposed data completion method quantitatively, we also compared the classification results based on the true and the predicted PET images. In addition, we report the classification results based on KNN and Zero methods. The AUC values of the three classification tasks based on true PET images and predicted images by three methods are given in Table 1.

We can observe from these results that the 3-D CNN model outperforms KNN and Zero methods significantly in all three classification tasks. These significant performance differences verify that our deep learning method successfully extracts highly nonlinear relationship between the MRI and PET images. We can also observe that the results of the 3-D CNN model is comparable with those of the true PET images. This demonstrates that our predicted PET images can potentially be used to improve the accuracy of disease diagnosis. Note that the classification performance reported here might be lower than those in the current literature on the ADNI data set because (1) we do not employ advanced feature extraction and classification methods on the true and completed data, and (2) the number of subjects used in the study is relatively small, since we used only these subjects with both MRI and PET.

3.3 Evaluation on All Subjects

To further evaluate the effectiveness of our proposed method, we report the prediction performance on all 830 subjects, where 398 subjects have both MRI and PET images, and the remaining 432 subjects have only MRI images. The 3-D CNN and other data completion methods were trained on the 398 subjects, and the trained models were used to complete the PET images of the remaining 432 subjects. The classification performance on all 830 subjects is reported in Table 2. Note that the comparison of classification performance based on true data is not applicable in this experiment, since 432 of 830 subjects did not have PET images.

We can observe that the 3-D CNN model outperforms KNN and Zero methods for all three tasks with three different combinations of PET and MRI modalities. This again demonstrates that the proposed 3-D CNN data completion method is more effective than the competing methods. We can also observe that the performance was improved when the MRI and PET

image features were combined. Overall, these experiments yielded insights on the power of the 3-D CNN model in completing missing neuroimaging data, thereby providing practical guidelines for employing multi-modality data even when some data modalities are missing. These results demonstrated that the estimated PET data could be used to improve the accuracy of disease diagnosis.

4 Conclusion and Future Work

We developed a 3-D CNN model for completing and integrating multi-modality neuroimaging data. This model takes one volumetric data modality as input and another modality as output. The nonlinear relationship between different data modalities is captured by a large number of trainable parameters in the network. We applied this model to predict the missing PET patterns from the MRI data. Results showed that the predicted PET data achieved similar classification performance as the true PET images. Additionally, our data completion method significantly outperformed the previous methods.

In this paper, we considered the CNN model for data completion. There are also other deep architectures that achieved promising performance on image-related tasks. It would be interesting to apply other deep models, such as the deep belief networks, for volumetric image data completion. In this work, we employed a CNN model with two hidden layers due to the high computational cost of training. We will explore ways of expediting the computation and design more complicated deep models in the future.

References

1. Weiner MW, et al. The Alzheimers Disease Neuroimaging Initiative: A review of papers published since its inception. Alzheimer's & Dementia. 2012; 8(suppl. 1):S1–S68.
2. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998; 86(11):2278–2324.
3. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet classification with deep convolutional neural networks. In: Pereira, F.; Burges, C.; Bottou, L.; Weinberger, K., editors. Advances in Neural Information Processing Systems. Vol. 25. Curran Associates, Inc.; 2012. p. 1106-1114.
4. Turaga SC, Murray JF, Jain V, Roth F, Helmstaedter M, Briggman K, Denk W, Seung HS. Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Computation. 2010; 22(2):511–538. [PubMed: 19922289]
5. Jain, V.; Seung, S. Natural image denoising with convolutional networks. In: Koller, D.; Schuurmans, D.; Bengio, Y.; Bottou, L., editors. Advances in Neural Information Processing Systems. Vol. 21. Curran Associates, Inc.; 2009. p. 769-776.
6. Cireşan, DC.; Giusti, A.; Gambardella, LM.; Schmidhuber, J. Mitosis detection in breast cancer histology images with deep neural networks. In: Mori, K.; Sakuma, I.; Sato, Y.; Barillot, C.; Navab, N., editors. MICCAI 2013, Part II. LNCS. Vol. 8150. Springer; Heidelberg: 2013. p. 411-418.
7. Ji S, Xu W, Yang M, Yu K. 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013; 35(1):221–231. [PubMed: 22392705]
8. Mutch, J.; Knoblich, U.; Poggio, T. Technical Report MIT-CSAIL-TR-2010-013 / CBCL-286. Massachusetts Institute of Technology; Cambridge, MA: Feb. 2010 CNS: a GPU-based framework for simulating cortically-organized networks.
9. Yuan L, Wang Y, Thompson PM, Narayan VA, Ye J. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. NeuroImage. 2012; 61(3):622–632. [PubMed: 22498655]

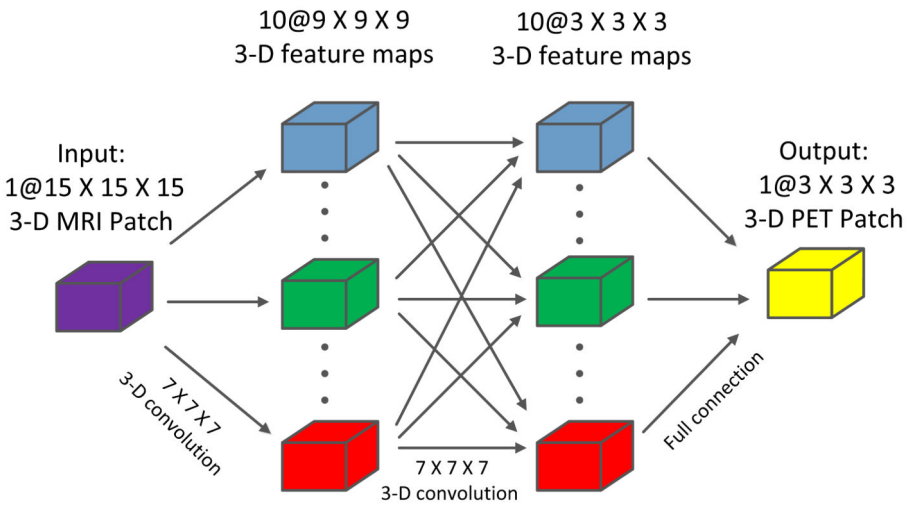

Fig. 1.

The 3-D CNN architecture for imaging data completion used in this work. There are 2 hidden layers between the input and output layers. Each of the hidden layers contains 10 feature maps. The total number of trainable parameters in this network is 37, 761.

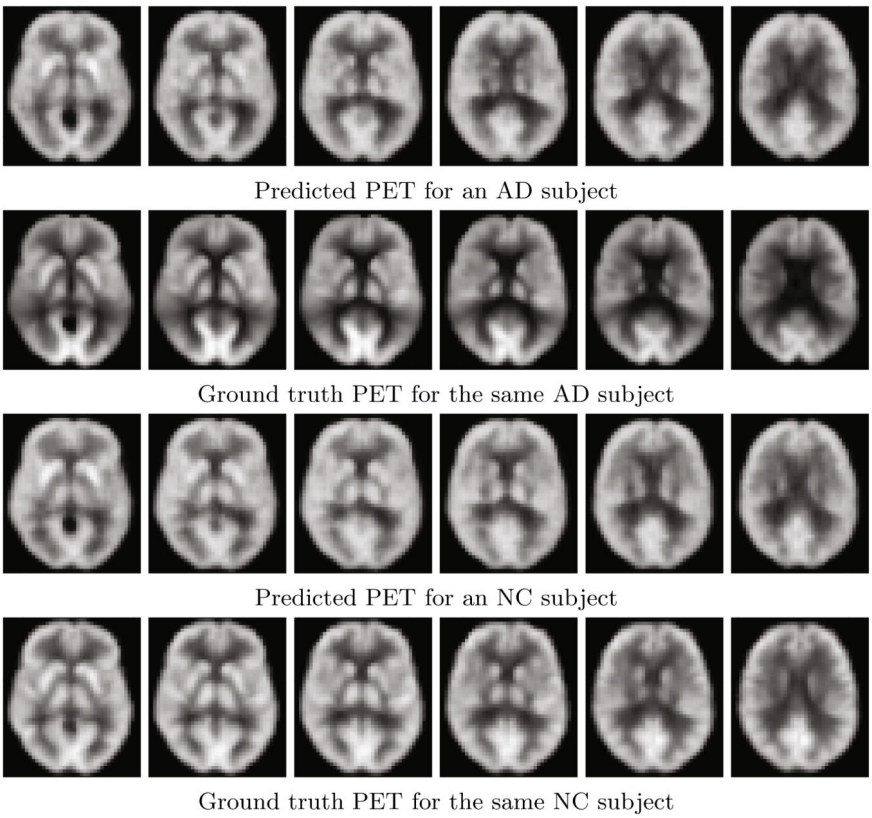

Fig. 2.

Comparison of the predicted and the ground truth PET images on two subjects. Each row corresponds to the data (either ground truth or predicted) of one subject, and each column corresponds to slice with the same brain position.

Table 1

Performance comparison of classification tasks using the true and the predicted PET data. The data set consists of 398 subjects having both MRI and PET images.

Tasks		MCI vs. NC	pMCI vs. sMCI	AD vs. NC
PET	True data	0.7014 ± 0.0212	0.6823 ± 0.0241	0.8982 ± 0.0224
	3-D CNN	0.6947 ± 0.0281	0.6804 ± 0.0267	0.8868 ± 0.0208
	KNN	0.6304 ± 0.0248	0.6278 ± 0.0326	0.7421 ± 0.0282
	Zero	0.6175 ± 0.0213	0.6124 ± 0.0243	0.6928 ± 0.0225

Table 2

Performance comparison of classification tasks using the true and the predicted PET images. All 830 subjects were used in this experiments, where subjects with no PET images were completed using three methods.

Tasks		MCI vs. NC	pMCI vs. sMCI	AD vs. NC
MRI		0.7439 ± 0.0329	0.7168 ± 0.0253	0.9192 ± 0.0188
PET	3-D CNN	0.7305 ± 0.0315	0.7029 ± 0.0245	0.8762 ± 0.0236
	KNN	0.6352 ± 0.0200	0.6133 ± 0.0346	0.7391 ± 0.0304
	Zero	0.6102 ± 0.0268	0.5924 ± 0.0331	0.7028 ± 0.0331
MRI + PET	3-D CNN	0.7621 ± 0.0205	0.7244 ± 0.0241	0.9287 ± 0.0207
	KNN	0.7231 ± 0.0214	0.6813 ± 0.0312	0.7691 ± 0.0213
	Zero	0.7217 ± 0.0290	0.6291 ± 0.0317	0.7003 ± 0.0162