Master’s thesis seminar

Unsupervised Representation Learning for Volume Rendering Diffusion MRI

Agajan Torayev

October 16, 2019

University of Bonn

Introduction

Introduction

Noninvasive medical imaging technologies such as MRI and diffusion MRI produce 3D arrays of data containing detailed representations of internal organs such as the brain of a human.

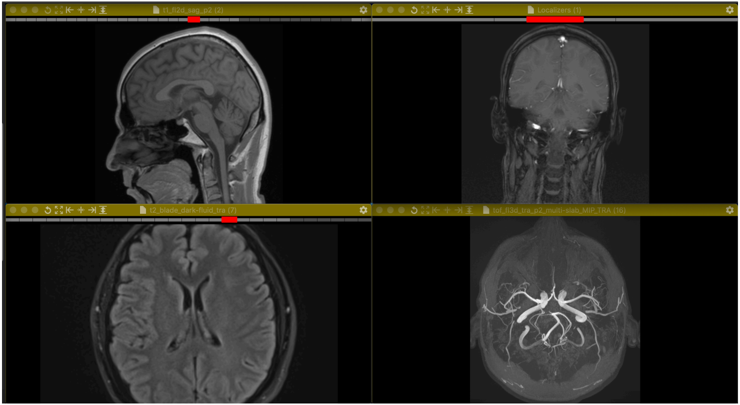
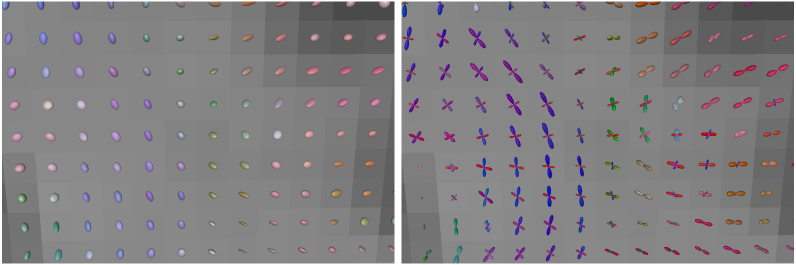

t1_fl2d_sag_p2 (2)
Localizers (1)
t2_blade_dark-fluid_tra (7)
tof_fl3d_tra_p2_multi-slab_MIP_TRA (16)

Figure 1: Brain MRI

Motivation

Data visualization techniques specialized in these types of data are necessary for interpretation and analysis:



(a) DTI glyphs

(b) ODF glyphs

Figure 2: Glyph based visualization

Motivation

One of the established technologies to visualize such type of data is direct volume rendering, a method for producing an image from a 3D array of sampled scalar data.

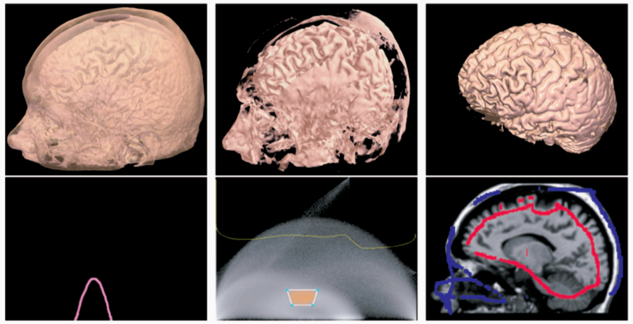


Figure 3: Left: Volume rendering of an MRI head data set with a 1D transfer function. Middle: Using a 2D transfer function. Right: Using a 10D classification function. Image and caption from [2].

Motivation

Given the complexity of diffusion MRI data, sketch-based transfer function design is the most promising strategy.

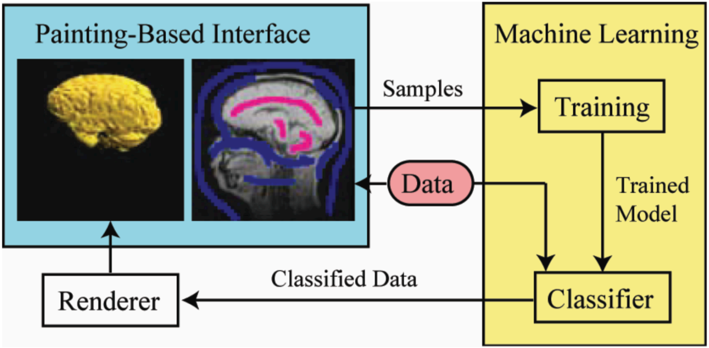


Figure 4: The painting-based intelligent user interface Image and caption from [2].

Motivation

The main goal of this thesis is to learn suitable representations of diffusion MRI that can be used to design transfer functions in direct volume rendering.

Evaluation Framework

Data

To train and evaluate the proposed methods, we used subject 784565 from the Human Connectome Project (HCP) [3].

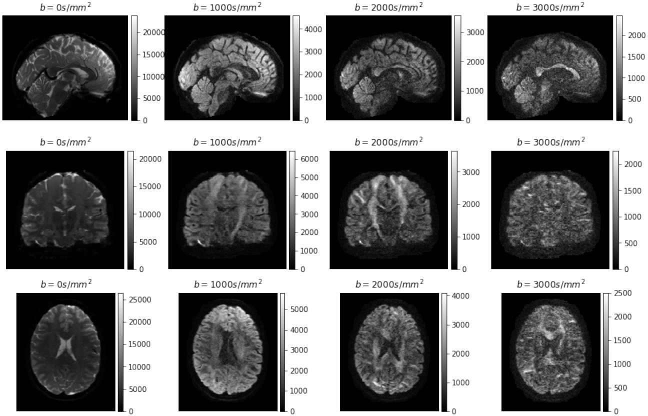


Figure 5: HCP subject 784565

Reference Segmentations

As reference segmentations, we used a subset of high-quality white matter reference tracts proposed by [4]: Cingulum (CG left, right), Corticospinal tract (CST left, right), Fornix (FX left, right), Corpus Callosum (CC all).

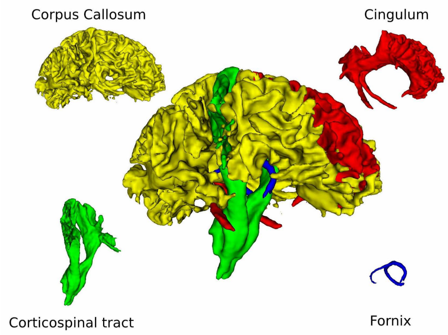


Figure 6: Segmentation of the HCP subject 784565.

Classifier

We can treat the task of tract segmentation as a dense classification problem where the goal is to assign labels to every single voxel.

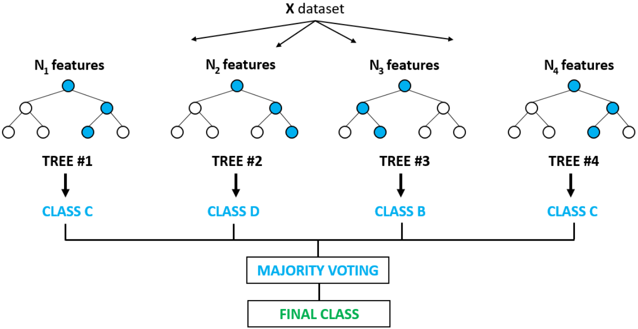


Figure 7: Random Forest Classifier.

*Image from https://www.globalsoftwaresupport.com/random-forest-classifier/

Dataset

The dataset for the classifier is given in the form of:

$$(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n) \text{ where } \mathbf{x}_i \in \mathbb{R}^n, \mathbf{y}_i \in \{0, 1\}^c \tag{1}$$

Here $\mathbf{x}_i$ represents an n-dimensional feature vector for the i^{th} voxel in a dataset, and $\mathbf{y}_i$ represents a 5-dimensional labels vector with value 1 in corresponding class dimension: for example $\mathbf{y}_i = [1, 0, 1, 0, 0]$ means that $\mathbf{x}_i$ at the same time belongs to the class "other" and the class CST.

Dataset

To train the classifier, we considered voxels in three center slices in each of the three orientations, i.e., slice 72 in sagittal orientation, slice 87 in coronal orientation and slice 72 in axial orientation. To test the classifier, we used two datasets:

1. **Test set 1**: Voxels in neighboring slices to the training slices, i.e., voxels in slice 71 in sagittal orientation, in slice 86 in coronal orientation and in slice 71 in axial orientation.
2. **Test set 2**: Voxels in the full brain volume inside the brain region.

Evaluation Metric

$$DICE = \frac{2|A \cap B|}{|A| + |B|} = \frac{2TP}{2TP + FP + FN} = F_1 \tag{2}$$

We calculate the *DICE* score for all five classes (including “other” class) separately but report the average of *DICE* scores for four tracts.

Establishing a Baseline

Baseline

To our knowledge, this is the first attempt to accomplish this goal. Thus, we established our baseline for this task.

Raw Diffusion Signal

For every voxel coordinate, we considered the measurements in time dimension as features for the classifier.

	CG	CST	FX	CC	Avg.
Test set 1	0.36896	0.73164	0.52336	0.78831	0.60307
Test set 2	0.20643	0.48117	0.29109	0.68683	0.41638

Table 1: The classification results for the raw diffusion signal features. For the Random Forest classifier we found that `min_samples_leaf` $= 8$ gives the best results.

SHORE

SHORE basis functions enable to reconstruct diffusion signals accurately with a smaller number of measurements [1].

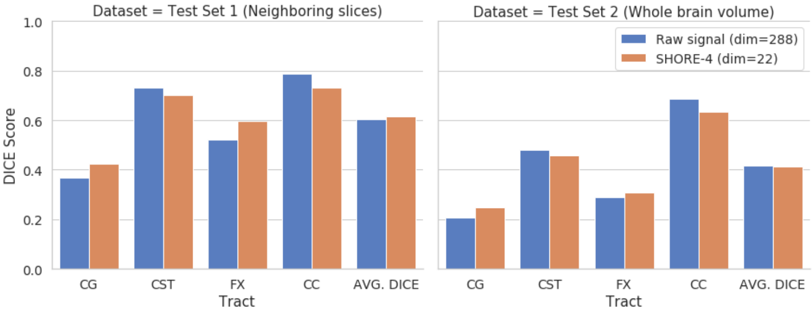


Figure 8: The classification results for the raw diffusion signal features and SHORE-4 features. For the average DICE score SHORE-4 yields very close results to the raw diffusion signal features.

PCA

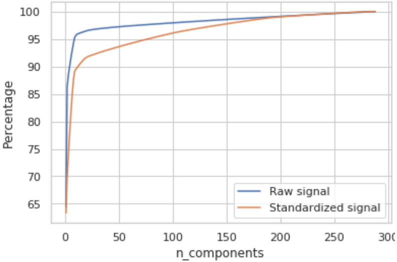


Figure 9: The analysis of the first k principal components. The first $k = 10$ principal components explain the 95.86% of variance.

PCA results

	CG	CST	FX	CC	Avg.
Test set 1	0.62128	0.74638	0.62992	0.7944	0.698
Test set 2	0.36875	0.4782	0.3413	0.68477	0.46825

Table 2: The classification results for the PCA ($k = 10$) features performed on raw diffusion signal without standardization.

Voxel coordinates

We found that concatenating spatial voxel positions with voxel features improves the classification results.

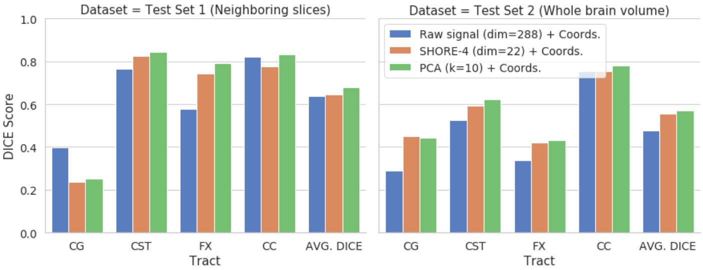


Figure 10: Baseline features are concatenated with spatial voxel coordinates. This process adds an extra three dimensions to the feature vector.

Autoencoders

Autoencoders

Autoencoder is a usual neural network which is trained to reconstruct input from a learned representation.

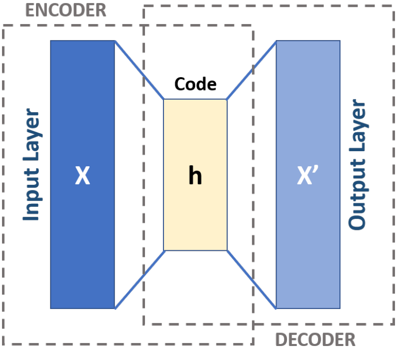


Figure 11: Autoencoder

*Image from https://en.wikipedia.org/wiki/Autoencoder

Hidden Dimension

	$h = 8$	$h = 10$	$h = 17$	$h = 22$	$h = 100$
Test set 1	0.62813	0.64340	**0.65615**	0.64366	0.56492
Test set 2	**0.44328**	0.43808	0.43932	0.43275	0.41002

Table 3: Classification results with different dimensional feature vectors taken from the bottleneck layer of the single hidden layer autoencoder.

Convolutional autoencoders

In convolutional autoencoders the affine transformations in the encoder and the decoder parts of the network are replaced with convolution operations.

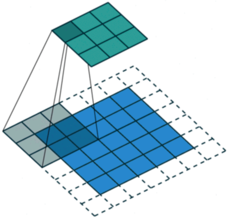
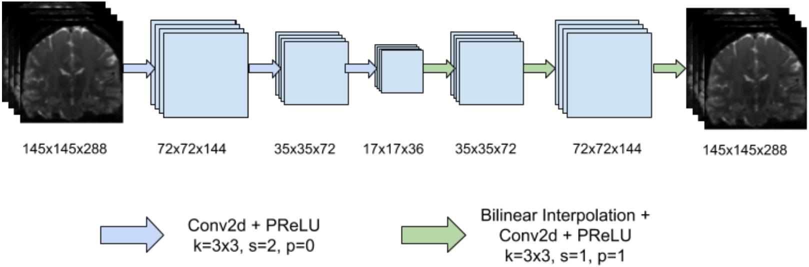


Figure 12: Convolutional operation

*Image from https://github.com/vdumoulin/conv_arithmetic

U-Net inspired autoencoder



Conv2d + PReLU
k=3x3, s=2, p=0

Bilinear Interpolation +
Conv2d + PReLU
k=3x3, s=1, p=1

Figure 13: U-Net inspired deep convolutional architecture. After the last convolutional layer PReLU activation function is not used.

U-Net inspired autoencoder

	sigmoid	tanh	ReLU	PReLU
Test set 1	0.08298	0.65361	0.43386	**0.72411**
Test set 2	0.08509	0.31259	0.29406	**0.38558**

Table 4: The effect of activation functions on the classification results in U-Net inspired deep convolutional autoencoder.

	Transpose Convolutions	Bilinear Interpolation
Test set 1	0.71096	**0.72411**
Test set 2	0.37973	**0.38558**

Table 5: The effect of upsampling method on the classification results in U-Net inspired deep convolutional autoencoder.

Single hidden layer convolution autoencoder

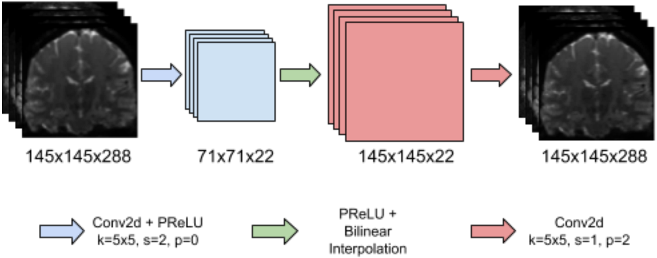


Figure 14: Single hidden layer convolutional autoencoder.

Single hidden layer convolution autoencoder

	$k = 3, h = 10$	$k = 3, h = 22$	$k = 5, h = 10$	$k = 5, h = 22$
Test set 1	0.64873	0.72191	0.70449	**0.74094**
Test set 2	0.40113	0.45717	0.43613	**0.48528**

Table 6: The effect of kernel size and hidden dimension on the classification results in the single hidden layer convolutional autoencoder. For both test sets big kernel and high dimension gives the highest scores.

Deep convolutional autoencoder

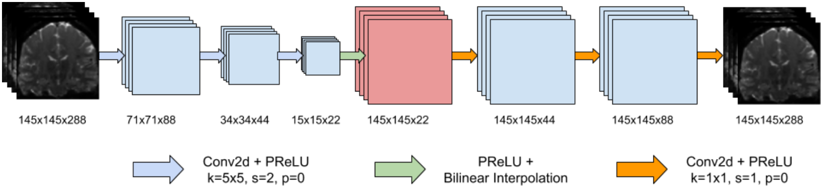


Figure 15: Deep Convolutional Architecture. After the last 1 × 1 convolutional layer PReLU activation function is not used.

Deep convolutional autoencoder

	Decoder with 1 × 1 kernels	Decoder with 5 × 5 kernels
Test set 1	**0.77257**	0.76698
Test set 2	**0.46936**	0.45664

Table 7: A decoder with 1 × 1 kernel size improves the classification results in deeep convolutional autoencoder. Also, using 1 × 1 kernel sizes in decoder reduces the number of parameters of the network. In this case the number of trainable parameters was reduced from 1510060 to 785644.

Features concatenation

	Simple AE ($h = 22$)	Deep Conv. AE	Features concat.
Test set 1	0.61524	0.77257	**0.82174**
Test set 2	0.43279	0.46936	**0.53390**

Table 8: The concatenation of features learned by simple hidden layer autoencoder with $h = 22$ and features learned by deep convolutional autoencoder with $h = 22$ channels in the bottleneck layer.

Multi-scale autoencoder

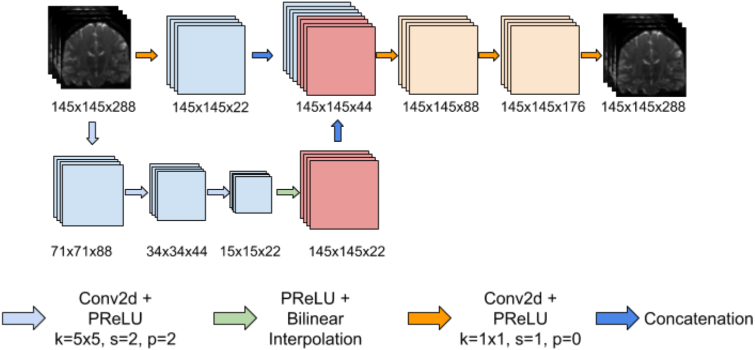


Figure 16: Multi-Scale Deep Convolutional Autoencoder. After the last 1×1 convolutional layer PReLU activation function is not used.

Multi-scale autoencoder

	CG	CST	FX	CC	Avg.
Test set 1	0.80521	0.86300	0.76154	0.86320	**0.82324**
Test set 2	0.46691	0.63187	0.42011	0.69962	**0.55463**

Table 9: The classification results obtained with features learned by multi-scale convolutional autoencoder.

Results

Quantitative results

	CG	CST	FX	CC	Avg.
PCA	0.62128	0.74638	0.62992	0.79440	0.69800
PCA + Coords.	0.25306	0.84300	**0.79245**	0.83400	0.68063
MS-AE	**0.80521**	0.86300	0.76154	0.86320	**0.82324**
MS-AE + Coords.	0.68852	**0.8725**	0.76364	**0.86725**	0.79798

Table 10: Test set 1.

	CG	CST	FX	CC	Avg.
PCA	0.36875	0.47820	0.34130	0.68477	0.46825
PCA + Coords.	0.44265	0.62371	**0.43164**	**0.77935**	0.56934
MS-AE	0.46691	0.63187	0.42011	0.69962	0.55463
MS-AE + Coords.	**0.47660**	**0.67267**	0.41928	0.72450	**0.57326**

Table 11: Test set 2.

Qualitative results

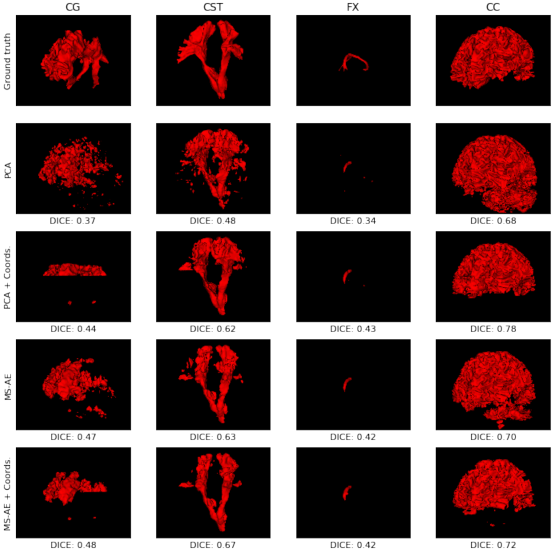


Figure 17: 3D segmentation results for the HCP subject 784565.

Reconstructions

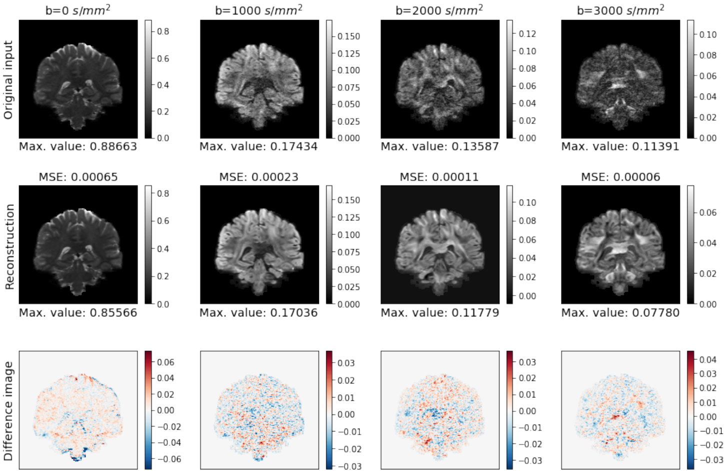


Figure 18: Images represent $t = 0, 1, 2, 3$ of the slice 72 in the coronal orientation where t is in the range $[0, 287]$

Learned Feature Maps

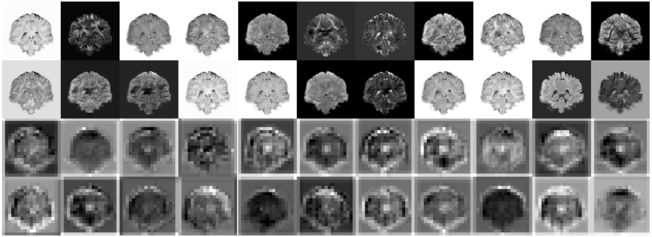


Figure 19: Features learned by multi-scale convolutional autoencoder for the slice 72 in coronal orientation of the HCP subject 784565.

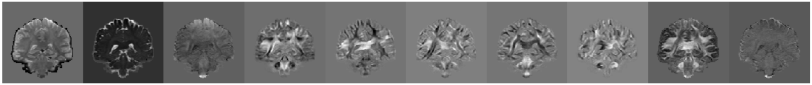
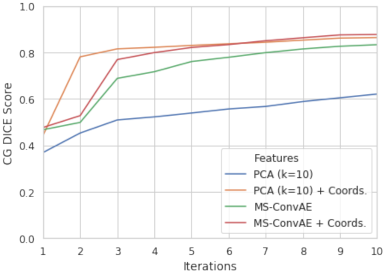


Figure 20: PCA ($k = 10$) feature maps for the slice 72 in coronal orientation of the HCP subject 784565.

Simulation

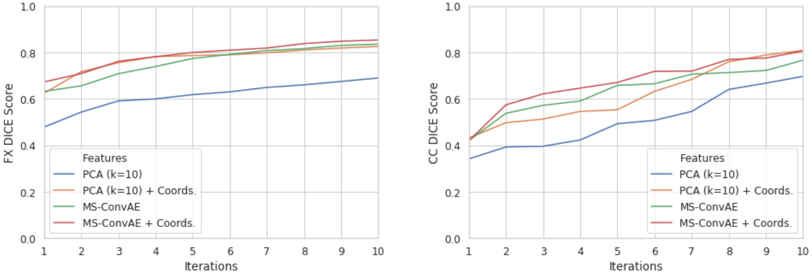


(a) CG tract

(b) CST tract

Figure 21: Simulation results for the HCP subject 784565.

Simulation



(a) FX tract

(b) CC tract

Figure 22: Simulation results for the HCP subject 784565.

Generalization

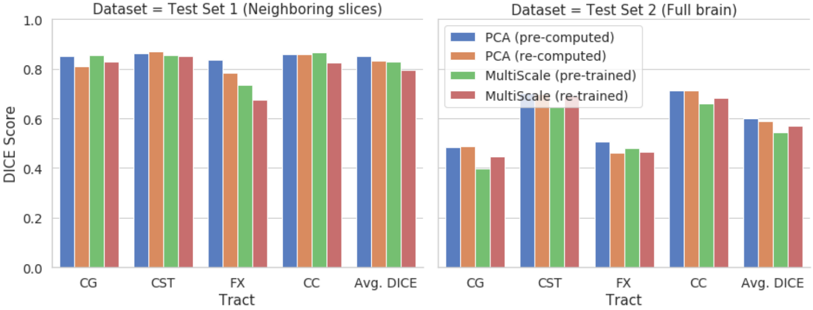


Figure 23: Classification results for the HCP subject 786569 (test subject.)

Generalization

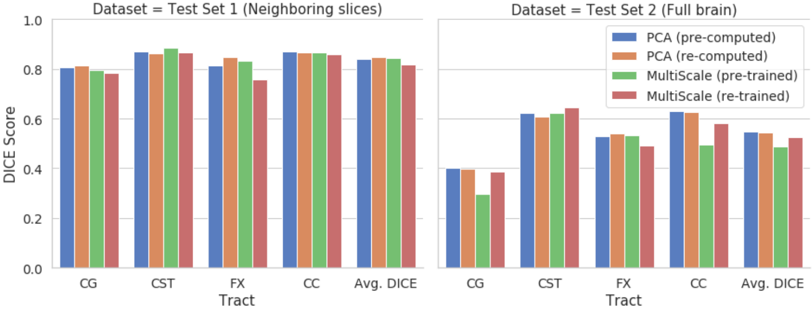
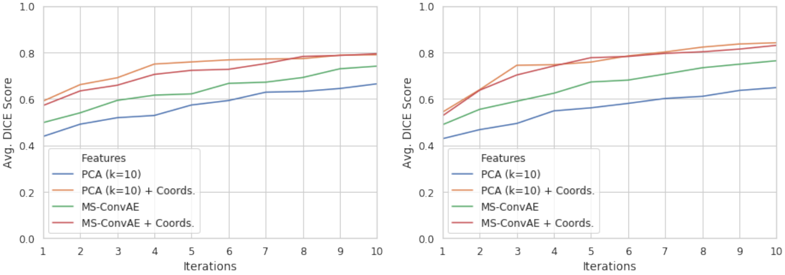


Figure 24: Classification results for the HCP subject 789373 (test subject.)

Generalization



(a) HCP subject 786569.

(b) HCP subject 789373.

Figure 25: Simulation results for the test HCP subjects.

Conclusion

Summary

Learned representations by convolutional autoencoder are superior to the PCA and SHORE features.

However, in our experiments, we found that concatenating spatial voxel coordinates to the features obtained by PCA makes them very strong baseline features.

Regarding the generalization of the autoencoder based methods, we found that they generalize relatively good to unseen data.

Questions?

Dataset Statistics

	Other class	CG	CST	FX	CC	**Total**
Train set	21728	1080	2244	141	6311	**31504**
Test set 1	22156	1058	2124	150	5955	**31443**
Test set 2	644241	42443	27212	1183	173114	**888193**

Table 12: The number of examples in each dataset for HCP subject 784565. The final dataset sizes for the training, Test 1 and Test 2 sets will be **29831**, **29940** and **844350** examples respectively. This happens because of overlapping labels, i.e., each voxel can belong to more than one label.

SHORE Coefficients

To obtain a different number of features, we experimented with radial order parameter of the SHORE basis functions.

	$\mathtt{r} = 2$	$\mathtt{r} = 4$	$\mathtt{r} = 6$
Test set 1	0.57137	**0.6133**	0.51284
Test set 2	0.36202	**0.41308**	0.36895

Table 13: The effect of $\mathtt{r}$ parameter to the classification results. Features are 7, 22, and 50-dimensional vectors respectively.

Activation Functions

	sigmoid	tanh	ReLU	PReLU
Test set 1	0.62303	**0.64340**	0.50134	0.63019
Test set 2	0.41966	**0.43808**	0.33401	0.43417

Table 14: The effect of activation functions on the classification results for the single hidden layer autoencoder. The latent representation is a 10-dimensional feature vector.

References i

S. Merlet and R. Deriche.
Continuous diffusion signal, eap and odf estimation via compressive sensing in diffusion mri.
Medical image analysis, 17(5):556–572, 2013.

F.-Y. Tzeng, E. B. Lum, and K.-L. Ma.
An intelligent system approach to higher-dimensional classification of volume data.
IEEE Transactions on visualization and computer graphics, 11(3):273–284, 2005.

D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, et al.
The wu-minn human connectome project: an overview.
Neuroimage, 80:62–79, 2013.

References ii

J. Wasserthal, P. Neher, and K. H. Maier-Hein.
Tractseg-fast and accurate white matter tract segmentation.
NeuroImage, 183:239–253, 2018.