VISUAL ANALYTICS AND INTERACTIVE MACHINE LEARNING FOR HUMAN BRAIN DATA

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Huang Li

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2019

Purdue University

Indianapolis, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Shiaofen Fang, Chair

Department of Computer and Information Science

Dr. Li Shen

Department of Computer and Information Science

Dr. Snehasis Mukhopadhyay

Department of Computer and Information Science

Approved by:

Dr. Shiaofen Fang

Department of Computer and Information Science

ACKNOWLEDGMENTS

I would like to express my special thanks to my advisor Prof. Shiaofen Fang and Prof. Shen Li for funding my research and guiding me during my Ph.D. program.

For the medical data in my research, I would like to thank the neuroscientists from IU Health Neuroscience Center including Prof. Andrew J. Saykin, Prof. Olaf Sporns, Prof. Joaquín Goñi Cortés and Prof. Yu-Chien Wu.

I would also like to thank Prof. Hongyang Chao of Sun Yat-sen University for the enlightenment of doing research during my Master program and further recommending me to pursuing my Ph.D. degree.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Li, Huang, Ph.D., Purdue University, August 2019. Visual Analytics and Interactive Machine Learning for Human Brain Data. Major Professor: Shiaofen Fang.

This study mainly focuses on applying visualization techniques on human brain data for data exploration, quality control, and hypothesis discovery. It mainly consists of two parts: multi-modal data visualization and interactive machine learning.

For multi-modal data visualization, a major challenge is how to integrate structural, functional and connectivity data to form a comprehensive visual context. We develop a new integrated visualization solution for brain imaging data by combining scientific and information visualization techniques within the context of the same anatomic structure. In this study, new surface texture techniques are developed to map non-spatial attributes onto both 3D brain surfaces and a planar volume map which is generated by the proposed volume rendering technique, Spherical Volume Rendering. Two types of non-spatial information are represented: (1) time-series data from resting-state functional MRI measuring brain activation; (2) network properties derived from structural connectivity data for different groups of subjects, which may help guide the detection of differentiation features. Through visual exploration, this integrated solution can help identify brain regions with highly correlated functional activations as well as their activation patterns. Visual detection of differentiation features can also potentially discover image based phenotypic biomarkers for brain diseases.

For interactive machine learning, nowadays machine learning algorithms usually require a large volume of data to train the algorithm-specific models, with little or no user feedback during the model building process. Such a big data based automatic learning strategy is sometimes unrealistic for applications where data collection or processing is very expensive or difficult. Furthermore, expert knowledge can be very valuable in the model building process in some fields such as biomedical sciences. In this study, we propose a new visual analytics approach to interactive machine learning. In this approach, multi-dimensional data visualization techniques are employed to facilitate user interactions with the machine learning process. This allows dynamic user feedback in different forms, such as data selection, data labeling, and data correction, to enhance the efficiency of model building. In particular, this approach can significantly reduce the amount of data required for training an accurate model, and therefore can be highly impactful for applications where large amount of data is hard to obtain. The proposed approach is tested on two application problems: the handwriting recognition (classification) problem and the human cognitive score prediction (regression) problem. Both experiments show that visualization supported interactive machine learning can achieve the same accuracy as an automatic process can with much smaller training data sets.

1. INTRODUCTION

Human brain data including structural-MRI, function-MRI and diffusion MRI [1] hold great promise for a systematic characterization of human brain connectivity and its relationship with cognition and behavior. This study mainly focus on applying visualization techniques on human brain data for data exploration, quality control, and hypothesis discovery.

The analysis of human brain data faces two major challenges:

1. How to seamlessly integrate computational methods with human knowledge and how to translate this into user-friendly, interactive software tools that optimally combines human expertise and machine intelligence to enable novel contextually meaningful discoveries. Both challenges require the development of highly interactive and comprehensive visualization tools that can guide researchers through a complex sea of data and information for knowledge discovery.

2. How to incorporate modern machine learning methods such as neural networks and support vector machines that use data to build computational models that are representations of nonlinear surfaces in high dimensional space. The trained models can then be used for analysis tasks such as classifications, regressions and predictions. Recent progress in deep learning has further empowered machine learning as an effective approach to a large set of big data analysis problems. As an automatic method, machine learning algorithms act mostly as a black box, i.e. the users have very little information about how and why the algorithm work or fail. The underlying machine learning models are also designed primarily for the convenience of learning from data, but they are not easy for the users to understand or interact with.

To address these challenges, this study breaks down to mainly two parts: the first part is multi-modal data visualization which focus on using advance techniques to give users an integrate view various the human brain data; the second part is about interactive machine learning which aims to provide a mechanism through visualization to allow users to understand and interact with the learning process.

For the first part, scientific visualization has traditionally been playing a role of visually interpreting and displaying complex scientific data, such as medical image data, to reveal structural and material details so as to help the understanding of the scientific phenomena. Example studies include diffusion tensor imaging (DTI) fiber tract visualization [2–7], network visualization [8–11], and multi-modal data visualization [12–14]. In this context, recent development in information visualization provides new ways to visualize non-structural attributes or in-depth analysis data, such as graph/network visualization and time-series data visualization. These, however, are usually separate visual representations away from the anatomic structures.

In order to maximize human cognitive abilities during visual exploration, this study proposes to integrate the visual representations of the connectome network attributes onto the surfaces of the anatomical structures of human brain. Multiple visual encoding schemes, combined with various interactive visualization tools, can provide an effective and dynamic data exploration environment for neuroscientists to better identify patterns, trends and markers. In addition, we develop a spherical volume rendering (SVR) algorithm using omni-directional ray casting and information encoded texture mapping. It provides a single 2D map of the entire rendered volume to provide better support for global visual evaluation and feature selection for analysis purpose.

Our primary contributions in the first part of this study include:

1. Development of a method to represent rich attribute information using information encoded textures.

2. Development of a new spherical volume rendering (SVR) technique that can generate a complete and camera-invariant view (volume map) of the entire structure.

3. Application of this approach to human brain visualization. Our experiments show great potential that this approach can be very useful in the analysis of neuroimaging data.

As to the second part, visualization, particularly multi-dimensional data visualization, has been playing an increasing important role in data mining and data analytics. This transformation of visualization from data viewing to being an integrated part of the analysis process led to the birth of the field of visual analytics [15]. In visual analytics, carefully designed visualization processes can effectively decode the insight of the data through visual transformations and interactive exploration. Many successful applications of visual analytics have been published in recent years, ranging from bioinformatics and medicine to engineering and social science. These success examples demonstrate that visualization is a powerful tool in data analytics that needs to be seriously considered in any big data application. On the other hand, automatic data mining and data analytics have made tremendous progress in the past decade. Machine learning, particularly deep learning, has become the mainstream analytics method in most big data analysis problems. The effective integration of visualization and machine learning/data mining is a new challenge in big data research.

Interactive machine learning aims to provide a mechanism through visualization to allow users to understand and interact with the learning process [16]. It has several important potential benefits.

1. Understanding. It is often difficult to improve the efficiency and performance of the algorithms without a clear understanding of how and why the different components work in machine learning algorithms. It is even more so in deep learning where there are large number of layers and interconnected components.

2. Knowledge Input. Human knowledge input can significantly improve the performance of machine learning algorithms, particularly in areas involving professional expertise such as medicine, science and engineering. Human instinct from visual perception can also out-perform computer algorithms. Hence, it is important to develop a visualization supported user feedback platform to allow user input to the machine learning system in the form of feature selection, dimension reduction, parameter setting, or addition / revision of rules and associations.

3. Data Reduction. Automatic machine learning usually requires a large volume of data to train the underlying computational model. This strategy sometimes is not realistic for applications in which data collection, labeling or processing is very expensive or difficult (for example, in clinical trials). Interactive visualization of the machine learning process allows the user to iteratively select the most critical and useful subset of data to be added to the training process so that the model building process is more data efficient (Figure 1.1). This is also our primary focus in this paper.

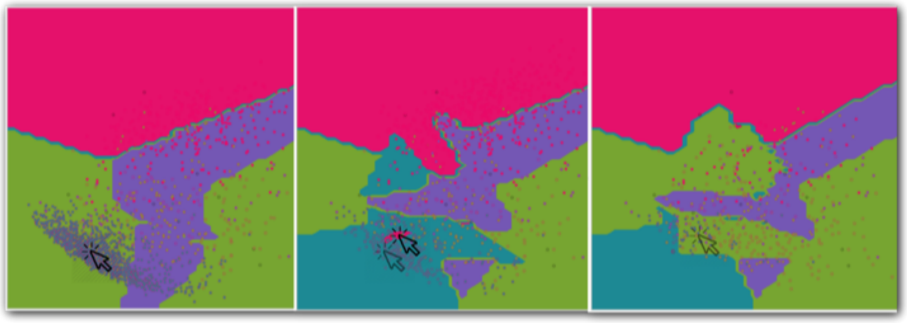


Fig. 1.1. An example of iterative model improvement by interactively adding new samples.

Our goal in the second part of this study is to develop a visualization supported user interaction platform in a machine learning environment such that the user can observe the evolution and performance of the internal structures of the model and provide feedback that may improve the efficiency of the algorithm or correct the direction of the model building process. Although the visualization platform we develop can be used to support understanding and knowledge input functions, we focus specifically in this paper on data reduction. In our approach, the interactive system will allow the user to identify potential areas (in some visual space) where additional data is needed to improve or correct the model (as shown in Figure 1.1). This way, only the necessary amount of data is used for learning a model. The aim is to solve a big data problem using a small data solution. In practice, this approach can not only save costs for data acquisitions / collections in applications such as clinical trials, medical analyses, and environmental studies, but also improve the efficiency and robustness of machine learning algorithms as the current somewhat brute-force approach (e.g. in deep learning) may not be necessary with smaller and higher quality data. To achieve this goal, we will need to overcome the following two challenges:

1. How to visualize the dynamics of a machine learning model is technically challenging. Previous works often depend on the specific machine learning algorithms. But in this paper, we will develop an approach and a general strategy that can be applied to most machine learning algorithms. In our test applications, support vector machines will be used as an example to demonstrate the effectiveness of this approach.

2. How to identify problematic areas from the visualization to revise the model, and how to efficiently and effectively provide user feedback to the algorithm are challenging. This is because machine learning features are often non-trivial properties of the data which cannot be easily used to pre-screen potential data collection target in real world applications.

In interactive machine learning, we will present a solution to these two challenges. Our approach will be tested on two practical applications with real world datasets.

In the following, we will first, in Section 2, discuss previous work related to human brain data visualization and interactive machine learning or other visual analytics solutions. In Section 3 we will describe the human brain data we used in this study. In Section 4, we will introduce the two parts of this study in detail, respectively. Conclusions and future work will be given in Section 5.

2. RELATED WORK

Human brain connectomics involves several different imaging modalities that require different visualization techniques. More importantly, multi-modal visualization techniques need to be developed to combine the multiple modalities and present both details and context for connectome related data analysis. Margulies, et al [3] provided an excellent overview of the various available visualization tools for brain anatomical and functional connectivity data. Some of these techniques are capable of carrying out multi-modal visualization involving magnetic resonance imaging (MRI), fiber-tracts as obtained from DTI and overlaying network connections. Various graphics rendering tools, along with special techniques such as edge bundling (to reduce clutter), have been applied to visualize DTI fiber tracts [2–5]. Due to tracking uncertainties in DTI fibers, these deterministic rendering can sometimes be misleading. Hence, rendering techniques for probabilistic DTI tractography have also been proposed [6, 7]. Several techniques have been developed to provide anatomical context around the DTI fiber tracts [12–14]. This typically require semi-transparent rendering with carefully defined transfer functions.

Multi-modal visualization is typically applied in the scientific visualization domain. The integration of information visualization and scientific visualization remains a challenge. In brain connectomics, connectome networks connectivity data are usually visualized as graphs. Graph visualization have been extensively studied in information visualization. For connectomics application, the networks can be either visualized as separate graphs, away from the anatomical context, but connected through interactive interfaces [8–11] or embedded into the brain anatomical context [17–19]. The embedded graphs, however, have their nodes constrained to their anatomical locations, and therefore do not need a separate graph layout process as in other graph visualization algorithms. Aside from embedded graphs, there has been little work in integrating more sophisticated information visualization, such as time-series data and multi-dimensional attributes, within the context of brain anatomical structures.

Many visualization techniques for time-series data have been developed in information visualization, such as time-series plot [20], spiral curves [21], and ThemeRiver [22], for non-spatial information and time-variant attributes. Several variations of ThemeRiver styled techniques have been applied in different time-series visualization applications, in particular in text visualization [23]. Depicting connectivity dynamics has been mostly done via traditional key-frame based approach [24, 25] or key frames combined with time-series plots [26, 27].

Texture-based visualization techniques have been widely used for vector field data, in particular, flow visualization. Typically, a grayscale texture is smeared in the direction of the vector field by a convolution filter, for example, the Line Integral Convolution (LIC), such that the texture reflects the properties of the vector field [28–30]. Similar techniques have also been applied to tensor fields [31, 32]].

As to volume datasets, volume rendering is a classic visualization technique. Both image-space and object-space volume rendering algorithms have been thoroughly studied in the past several decades. The typical image-space algorithm is ray casting, which was first proposed by Levoy [33]. Many improvements of ray casting have since been developed [34–37]. Splatting is the most common object-space approach. It directly projects voxels to the 2D screen to create screen footprints which can be blended to form composite images [38–42]. Hybrid approaches such as shearwrap algorithm [43] and GPU based algorithms provide significant speedup for interactive applications [44, 45]. Although iso-surfaces are typically extracted from volume data as polygon meshes [46], ray casting methods can also be applied towards volumetric isosurfacing [47, 48].

Although interactive machine learning has been previously proposed in the machine learning and AI communities [16,49], applying visualization and visual analytics principles in interactive machine learning has only been an active research topic in recent years. Most of the existing studies focus on using visualization for better understanding of the machine learning algorithms. There have also been some recent works on using visual analytics for improving the performance of machine learning algorithms through better feature selection or parameter setting.

While there have been many literatures on using interactive visualization to directly accomplish analysis tasks such as classification and regression [50, 51], we will focus mostly on approaches that deal with some machine learning models [52]. Previous works on using visualization to help understand the machine learning processes are usually designed for specific types of algorithms, for example, support vector machines, neural networks, and deep learning neural networks.

Neural Networks received the most attention due to its black box nature of the learning model and the complexity of its internal components. Multi-dimensional visualization techniques such as scatterplot matrix have been used to depict the relationships between different components of the neural networks [53, 54]. Typically, a learned component is represented as a higher dimensional point. The 2D projections of these points in either principal component analysis (PCA) spaces or a multi-dimensional scaling (MDS) space can better reveal the relationships of these components that are not easily understood, such as clusters and outliers. Several techniques have applied graph visualization techniques to visualize the topological structures of the neural networks [55–57]. Visual attributes of the graph can be used to represent various properties of the neural network models and processes.

Several recent studies tackle specific challenges in the visualization of deep neural networks due to the large number of components, connections and layers. In [58]. Liu et al. developed a visual analytics system, CNNVis, that helps machine learning experts understand deep convolutional neural networks by clustering the layers and neurons. Edge bundling is also used to reduce visual clutter. Techniques have also been developed to visualize the response of a deep neural network to a specific input in a real-time dynamic fashion [59, 60]. Observing the live activations that change in response to user input helps build valuable intuitions about how convnets work.

There are several literatures discussing visualizations roles in support vector machines. In [61], visualization methods are used to provide access to the distance measure of each data point to the optimal hyperplane as well as the distribution of distance values in the feature space. In [62], multi-dimensional scaling technique is used to project high-dimensional data points and their clusters onto a two-dimensional map preserving the topologies of the original clusters as much as possible to preserve their support vector models.

Visualization and visual analytics methods have been proposed for the performance analysis of machine learning algorithms in different applications [63–65]. Interactive methods have also been proposed to improve the performance of machine learning algorithms through feature selection and optimization of parameter settings. Some general discussions are given in [52] and [66]. In [67], a visual analytics system for machine learning support called Prospector is described. Prospector supports model interpretability and actionable insights, and provides diagnostic capabilities that communicate interactively how features affect the prediction. In [68], a multi-graph visualization method is proposed to select better features through an interactive process for the classification of brain networks. Other performance improvement methods include training sample selection and classifier tuning [69] and model manipulations by user knowledge [70–72].

The incremental visual data classification method proposed in [69] has some similarities conceptually to what we propose in this paper. In [69], neighbor joining tree is used to classify 2D image data. The model building process is done incrementally by adding additional images that are visually similar to the test samples that were misclassified. This approach puts a very heavy burden on the user as finding similar images by the user from a large image database or other sources is difficult and time-consuming. Our approach, on the other hand, is a more general framework that works for all machine learning algorithms and all data types. It is designed to allow incremental addition of training data with any user defined characteristics (attributes) that are easy to identify and collect.

There have been a number of previous works on using visualization to help understand the machine learning processes. Neural Networks received the most attention due to its black box nature of the learning model and the complexity of its internal components. Multi-dimensional visualization techniques such as scatterplot matrix have been used to depict the relationships between different components of the neural networks [73, 74]. Typically, a learned component is represented as a higher dimensional point. The 2D projections of these points in either principal component analysis (PCA) spaces or a multi-dimensional scaling (MDS) space can better reveal the relationships of these components that are not easily understood, such as clusters and outliers. Several techniques have applied graph visualization techniques to visualize the topological structures of the neural networks [75–77]. Visual attributes of the graph can be used to represent various properties of the neural network models and processes.

Several recent studies tackle specific challenges in the visualization of deep neural networks due to the large number of components, connections and layers. In [78]. Liu et al. developed a visual analytics system, CNNVis, that helps machine learning experts understand deep convolutional neural networks by clustering the layers and neurons. Edge bundling is also used to reduce visual clutter. Techniques have also been developed to visualize the response of a deep neural network to a specific input in a real-time dynamic fashion [79, 80]. Observing the live activations that change in response to user input helps build valuable intuitions about how convnets work. The DQNViz system [81] provides a visual analytics environment for the understanding of a deep reinforcement learning model. GAN Lab system [82] uses visualization to help non-expert users to learn how a Deep Generative Model works.

Several studies focus on visualizing the features captured by deep neural network. Class Activation Mapping (CAM) [83] was proposed by Zhou et al as a method for identifying discriminative locations used by the convolutional layers in deep learning model without any fully-connected layer. By extending CAM with gradient, Selvaraju et al proposed Grad-CAM [84] which works with fully-connected layers. But these methods did not reveal how the nodes in the hidden layer capture features from those discriminative locations and aggregate them together in the feed-forward process, which is important for users to make sense of the deep learning model.

Visualization and visual analytics methods have been proposed for the performance analysis of machine learning algorithms in different applications [85–87]. Interactive methods have also been proposed to improve the performance of machine learning algorithms through feature selection and optimization of parameter settings. Some general discussions are given in [88] and [89]. In [90], a visual analytics system for machine learning support called Prospector is described. Prospector supports model interpretability and actionable insights, and provides diagnostic capabilities that communicate interactively how features affect the prediction. In [91], a multi-graph visualization method is proposed to select better features through an interactive process for the classification of brain networks. Other performance improvement methods include training sample selection and classifier tuning [92] and model manipulations by user knowledge [93–95].

More generic visualization methods for general machine learning models have been developed in recent years. The Manifold system [96] provides a generic framework that does not rely on or access the internal logic of the model and solely observes the input and output. An ontology, VIS4ML, is proposed in [97] for VA-Assisted machine learning. [98], A generic visualization method for machine learning model is proposed to help select the optimal set of sample input data.

Compared to the existing methods on visualization in machine learning and deep learning applications, our approach emphasizes building association relationships between hidden layer nodes and the brain image features (phenotypes) such that user can observe and interact directly with the complex anatomical structures of the brain during the deep learning process. This work is also a good example of how to integrate scientific visualization and information visualization techniques in a deep learning visual analytics platform.

3. BRAIN IMAGING DATA AND CONNECTOME CONSTRUCTION

We first describe the MRI and DTI data used in this study, then present our methods for constructing connectome networks from the MRI and DTI data, and finally discuss the resting state functional MRI (fMRI) data used in our time-series visualization study.

The MRI and DTI data used in the preparation of this article were obtained from the Alzheimers Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership, led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial MRI, positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimers disease (AD). For up-to-date information, see www.adni-info.org.

We downloaded the baseline 3T MRI (SPGR) and DTI scans together with the corresponding clinical data of 134 ADNI participants, including 30 cognitively normal older adults without complaints (CN), 31 cognitively normal older adults with significant memory concerns (SMC), 15 early MCI (EMCI), 35 late MCI (LMCI), and 23 AD participants. In our multi-class disease classification experiment, we group these subjects into three categories: Healthy Control (HC, including both CN and SMC participants, N=61), MCI (including both EMCI and LMCI participants, N=50), and AD (N=23).

Using their MRI and DTI data, we constructed a structural connectivity network for each of the above 134 participants. Our processing pipeline is divided into three major steps described below: (1) Generation of regions of interest (ROIs), (2) DTI tractography, and (3) connectivity network construction.

(1) ROI Generation. Anatomical parcellation was performed on the high-resolution T1-weighted anatomical MRI scan. The parcellation is an automated operation on each subject to obtain 68 gyral-based ROIs, with 34 cortical ROIs in each hemisphere, using the FreeSurfer software package (http://freesurfer.net/). The Lausanne parcellation scheme [99] was applied to further subdivide these ROIs into smaller ROIs, so that brain networks at different scales (e.g., N_roi = 83, 129, 234, 463, or 1015 ROIs/nodes) could be constructed. The T1-weighted MRI image was registered to the low resolution b0 image of DTI data using the FLIRT toolbox in FSL, and the warping parameters were applied to the ROIs so that a new set of ROIs in the DTI image space were created. These new ROIs were used for constructing the structural network.

(2) DTI tractography. The DTI data were analyzed using FSL. Preprocessing included correction for motion and eddy current effects in DTI images. The processed images were then output to Diffusion Toolkit (http://trackvis.org/) for fiber tracking, using the streamline tractography algorithm called FACT (fiber assignment by continuous tracking). The FACT algorithm initializes tracks from many seed points and propagates these tracks along the vector of the largest principle axis within each voxel until certain termination criteria are met. In our study, stop angle threshold was set to 35 degree, which meant if the angle change between two voxels was greater than 35 degree, the tracking process stopped. A spline filtering was then applied to smooth the tracks.

(3) Network Construction. Nodes and edges are defined from the previous results in constructing the weighted, undirected network. The nodes are chosen to be N_roi ROIs obtained from Lausanne parcellation. The weight of the edge between each pair of nodes is defined as the density of the fibers connecting the pair, which is the number of tracks between two ROIs divided by the mean volume of two ROIs [100]. A fiber is considered to connect two ROIs if and only if its end points fall in two ROIs respectively. The weighted network can be described by a matrix. The rows

and columns correspond to the nodes, and the elements of the matrix correspond to the weights.

To demonstrate our visualization scheme for integrative exploration of the time-series of resting-state fMRI (rs-fMRI) data with brain anatomy, we employed an additional local (non-ADNI) subject, who was scanned in a Siemens PRISMA 3T scanner (Erlangen Germany). A T1-weighted sagittal MP-RAGE was obtained (TE =2.98 ms, TR partition = 2300ms, TI = 900ms, flip angle = 9, 128 slices with 111mmvoxels). A resting-state session of 10 minutes was also obtained. Subject was asked to stay still and awake, and to keep eyes closed. BOLD acquisition parameters were: TE = 29ms, TR = 1.25s, Flip angle = 79, 41 contiguous interleaved 2.5 mm axial slices, with in-plane resolution = 2.5 2.5 mm. BOLD time-series acquired were then processed according to the following steps (for details see [101]): mode 1000 normalization; z-scoring and detrending; regression of 18 detrended nuisance variables (6 motion regressors [X Y Z pitch jaw roll], average gray matter (GM), white matter (WM) and cerebral spinal fluid (CSF) signals, and all their derivatives computed as backwards difference); band-pass filter of 0.009 to 0.08 Hz using a zero-phase 2nd order Butterworth filter; spatial blurring using a Gaussian filter (FWHM=2mm); regression of the first 3 principal components of WM (mask eroded 3 times) and CSF (ventricles only, mask eroded 1 time). The Desikan-Killiany Atlas (68 cortical ROIs, as available in the Freesurfer software) was registered to the subject. The resulting processed BOLD time-series where then averaged for each ROI. Note that the Lausanne parcellation scheme (mentioned above) at the level of N_roi = 83 consists of the above 68 cortical ROIs together with the brain stem (as 1 ROI) and 14 subcortical ROIs. As a result, we will use 68 time series (one for each cortical ROI) in our time series visualization experiments.

4. METHODS AND RESULTS

4.1 Multi-modal Visualization

In this section, we propose a few information visualization methods. Using the VTK (www.vtk.org) C++ library, we have implemented and packaged these methods into a software tool named as BECA, standing for Brain Explorer for Connectomic Analysis. A prototype software is available at `http://www.iu.edu/~beca/`.

4.1.1 Visualizing Structural Connectivity Networks

3D visualization of a connectivity network within an anatomical structure can provide valuable insight and better understanding of the brain networks and their functions. In a brain network, we render nodes as ROI surfaces, which are generated using an iso-surface extraction algorithm from the MRI voxel sets of the ROIs. Drawing the network edges is, however, more challenging since straight edges will be buried inside the brain structures. We apply the cubic Bezier curves to draw curved edges above the brain structure. The four control points of each edge are defined by the centers of the ROI surfaces and the extension points from the centroid of the brain, as shown in Figure 4.1. Figure 4.2 shows visualization examples of a connectome network, along with the cortical surface, the ROIs, and the DTI fibers.

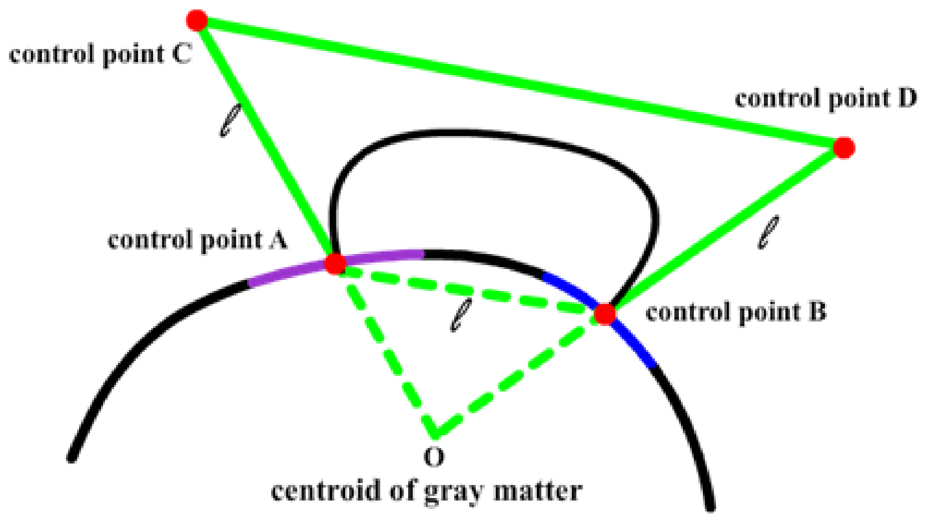


Fig. 4.1. Building a Bezier curve connecting two ROIs.

Brain connectivity networks obtained through the above pipeline can be further taken into complex network analysis. Network measures (e.g., node degree, betweenness, closeness) can be calculated from individuals or average of a population. Different measures may characterize different aspects of the brain connectivity [102]. In order to visualize these network attributes, we propose a surface texture based approach. The main idea is to take advantage of the available surface area of each ROI, and encode the attribute information in a texture image, and then texture-map this image to the ROI surface. Since the surface shape of each ROI (as a triangle mesh) is highly irregular, it becomes difficult to assign texture coordinates for mapping the texture images. We apply a simple projection plane technique. A projection plane of an ROI is defined as the plane with a normal vector that connects the center of the ROI surface and the centroid of the entire brain. The ROI surface can then be projected onto its projection plane, and the reverse projection defines the texture mapping process. Thus, we can define our attribute-encoded texture image on this project plane to depict a visual pattern on the ROI surface. Visually encoding attribute information onto a texture image is an effective way to represent multiple

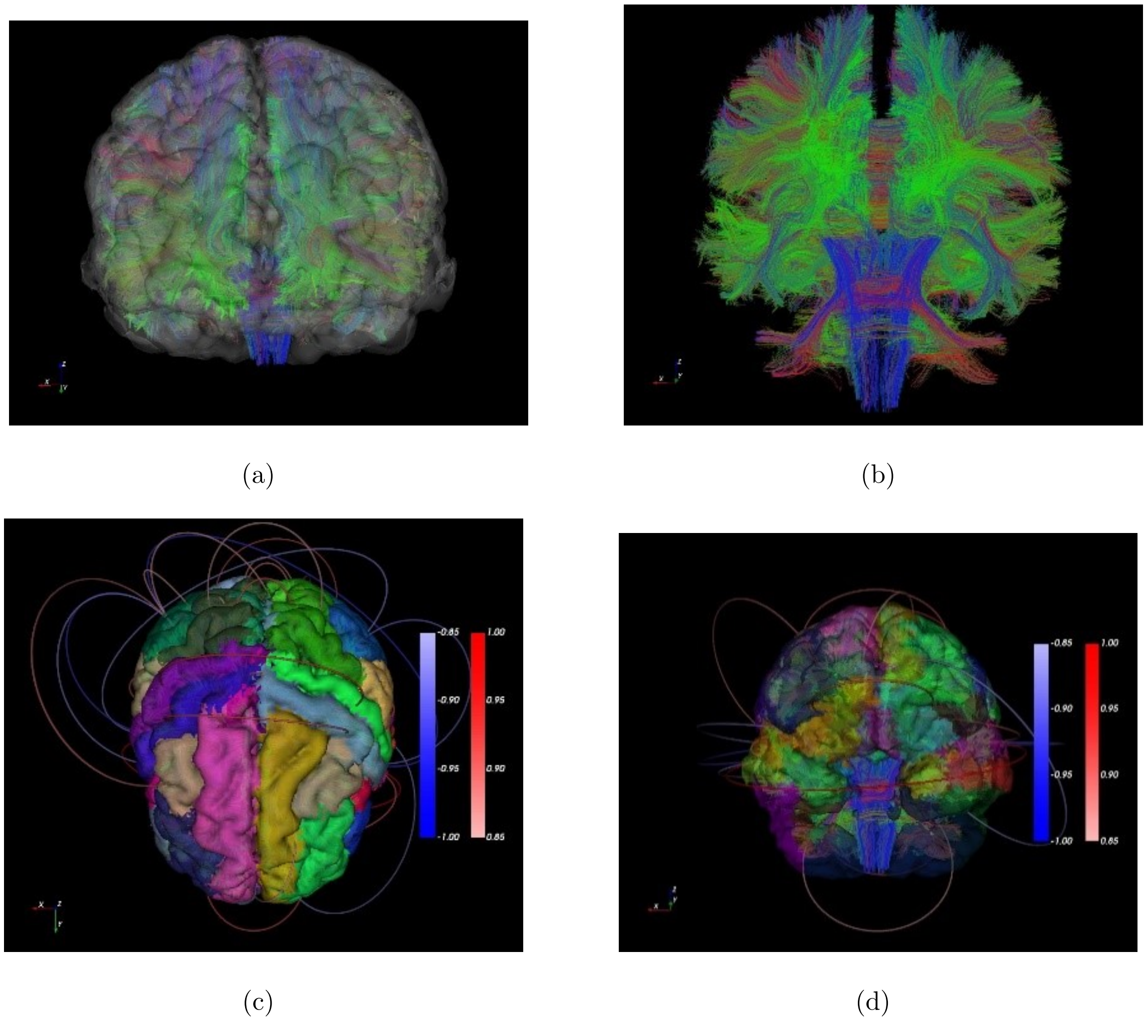


Fig. 4.2. (a) DTI fiber tracts; (b) MRI-ROIs and DTI fibers; (c,d) Network edges as Bezier curves (thresholded by edge intensity)

attributes or time-series attributes. Below we will demonstrate this idea in two different scenarios: Time-series data from rs-fMRI and multi-class disease classification.

4.1.2 Visualizing fMRI Data and Functional Connectivity

As a functional imaging method, rs-fMRI can measure interactions between ROIs when a subject is resting [103]. Resting brain activity is observed through changes in blood flow in the brain which can be measured using fMRI. The resting state approach is useful to explore the brain's functional organization and to examine if it is altered in neurological or psychiatric diseases. Brain activation levels in each ROI represent a time-series that can be analyzed to compute correlations between different ROIs. This correlation based network represents the functional connectivity networks and, analogously to structural connectivity, it may be represented as a square symmetric matrix.

Using the surface texture mapping approach, we need to first encode this time-series data on a 2D texture image. We propose an offset contour method to generate patterns of contours based on the boundary of each projected ROI. The offset contours are generated by offsetting the boundary curve toward the interior of the region, creating multiple offset boundary curves, as shown in Figure 4.3. There are several offset curve algorithms available in curve/surface modeling. Since in our application, the offset curves do not need to be very accurate, we opt to use a simple image erosion algorithm [104] directly on the 2D image of the map to generate the offset contours.

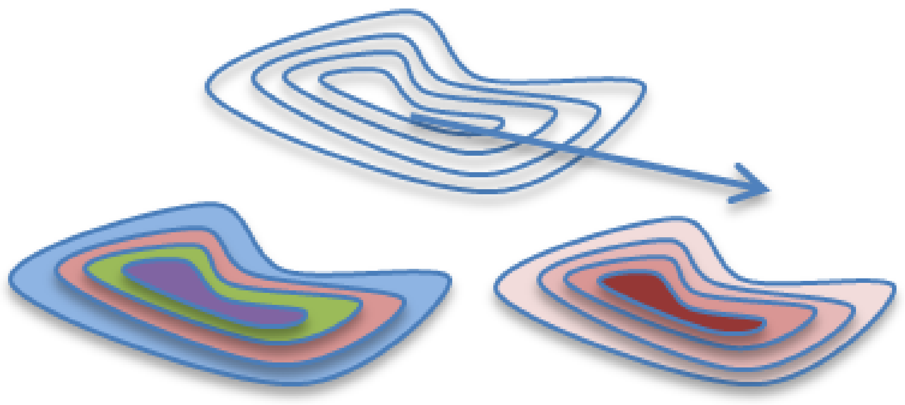


Fig. 4.3. Offset contours with different colors or different shades of the same color.

In time-series data visualization, the time dimension can be divided into multiple time intervals and represented by the offset contours. Varying shades of a color hue can be used to represent the attribute changes over time. Figure 4.4 shows the steps for constructing the contour-based texture. First, we map each ROI onto a projection plane perpendicular to the line connecting the centroid of the brain and the center of this ROI. The algorithm then iteratively erodes the mapped shape and assigns colors according to the activity level of this ROI at each time point. Lastly we overlay the eroded regions to generate a contour-based texture. We also apply a Gaussian filter to smooth the eroded texture image to generate more gradual changes of the activities over time. Figure 4.5 shows a few examples of the offset contours mapped to the ROIs. The original data has 632 time points, which will be divided evenly across the contours depending on the number of contours that can be fitted in to the available pixels within the projected ROI.

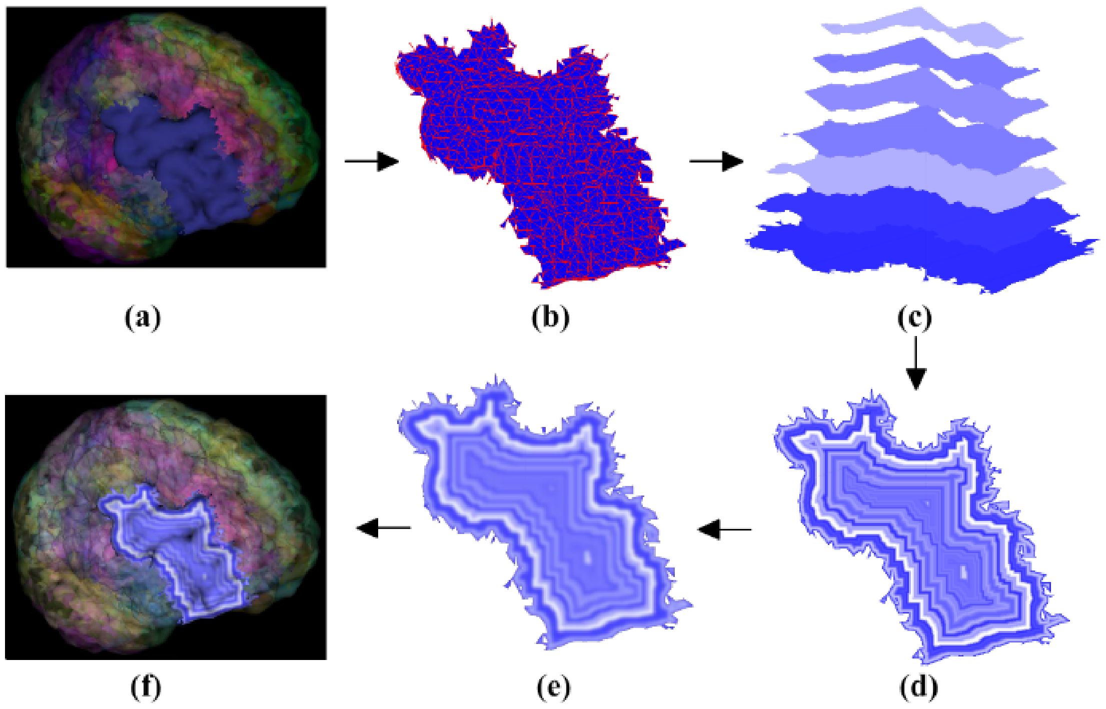


Fig. 4.4. (a) Original ROI. (b) ROI mapping. (c) Iterative erosion. (d) Overlaying. (e) Gaussian blurring. (f) Applying the texture.

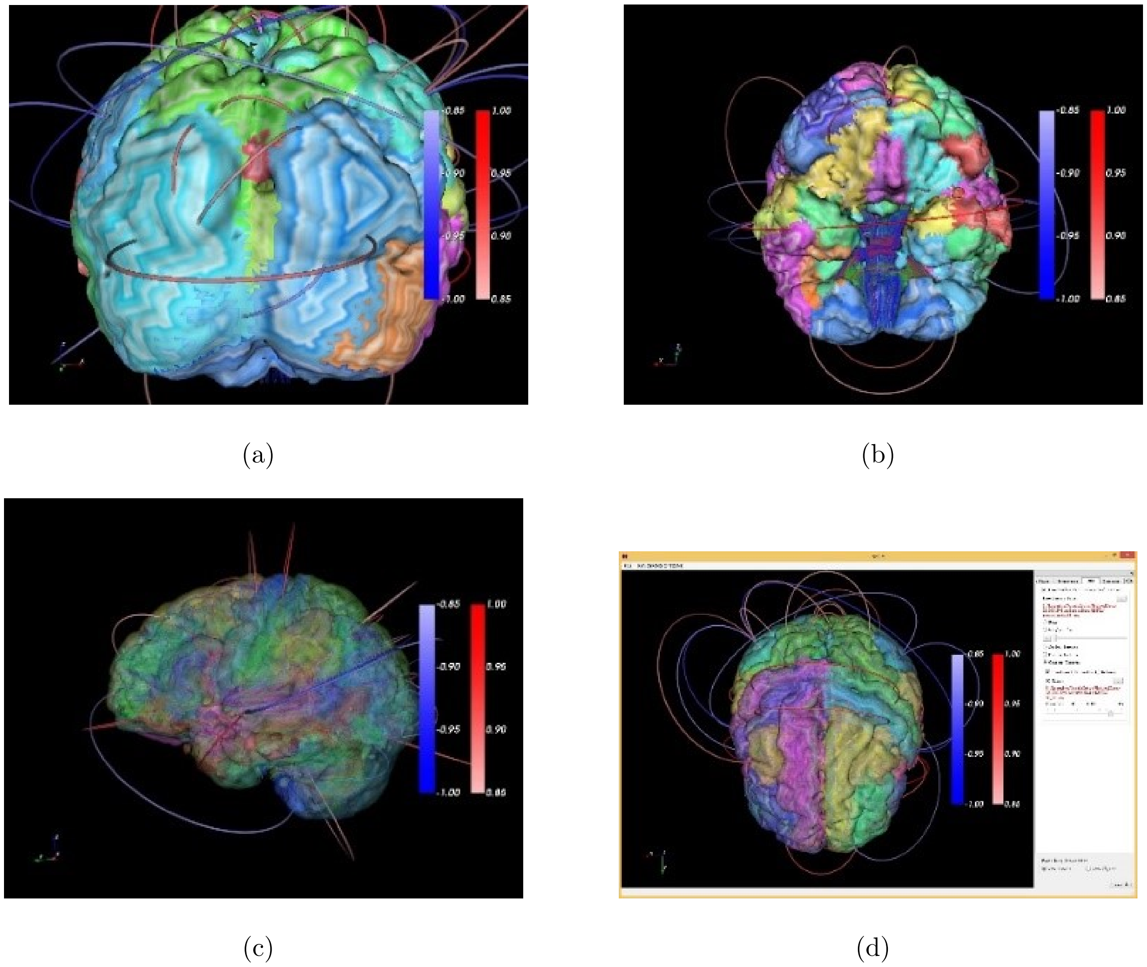


Fig. 4.5. Some examples of a connectome network with time-series data. Various transparencies are applied.

4.1.3 Visualizing Discriminative Patterns among Multiple Classes

In this case study, we performed the experiment on the ADNI cohort mentioned before, including 61 HC, 50 MCI and 23 AD participants. The goal is to generate intuitive visualization to provide cognitively intuitive evidence for discriminating ROIs that can separate subjects in different classes. This can be the first step of a diagnostic biomarker discovery process.

The goal of the visual encoding in this case is to generate a color pattern that can easily distinguish bias toward any of the three classes. To do so, we first assign a distinct color to each class. Various color patterns can be generated using different color blending and distribution methods. In our experiment, a noise pattern is applied with 3 colors representing the 3 classes. The same noise pattern approach can also accommodate more colors.

Since color blending is involved in a noise pattern, we choose to use an RYB color model, instead of the RGB model. This is because color mix using RYB model is more intuitive in a way that the mixed colors still carry the proper amount of color hues of the original color components. For example, Red and Yellow mix to form Orange, and Blue and Red mix to form Purple. Thus, RYB model can create color mixtures that more closely resemble the expectations of a viewer. Of course these RYB colors still need to be eventually converted into the RGB values for display. For the conversion between these two color models, we adopt the approach proposed in [105, 106], in which a color cube is used to model the relationship between RYB and RGB values. For each RYB color, its approximated RGB value can be computed by a trilinear interpolation in the RYB color cube.

We first construct noise patterns to create a random variation in color intensity, similar to the approach in [105]. Different color hues are used to represent the attributes in different classes of subjects. Any network measurement can be used for color mapping. In our experiment, we use the node degrees averaged across subjects in each class. A turbulence function [107] is used to generate the noise patterns of different frequencies (sizes of the sub-regions of the noise pattern). An example is shown in Figure 4.6, we blend RYB channels with weights 0.5, 0.25and 0.25 respectively. The blended texture is red-dominated with a little yellow and blue color.

Figure 4.7 shows some examples of the texture mapped views of the three classes: HC (Red), MCI (Yellow) and AD (Blue). The colors of the edges also represent the blended RYB color values, based on the average edge weights in the three classes. From the resulting images, we can identify a specific ROI that exhibits bias toward

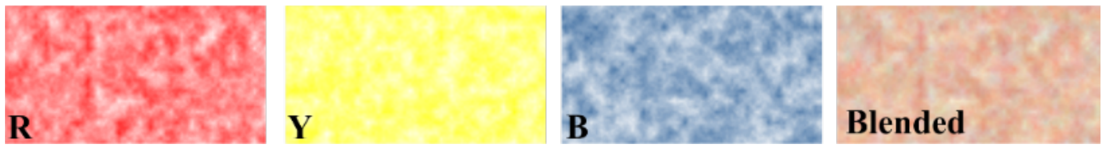


Fig. 4.6. Blending RYB channels with weights 0.5, 0.25 and 0.25.

one or two base colors. This can be a potential indication that this ROI may be a good candidate for further analysis as a potential imaging phenotypic biomarker.

4.1.4 Spherical Volume Rendering (SVR)

In previous sections, we mapped attributes onto the ROI surface. However, each rendering shows only one perspective, and subcortical structures remain unseen. Therefore, it does not provide an overall view of the complete structure. In this section, we develop a spherical volume rendering algorithm that provides a single 2D map of the entire brain volume to provide better support for global visual evaluation and feature selection for analysis purpose.

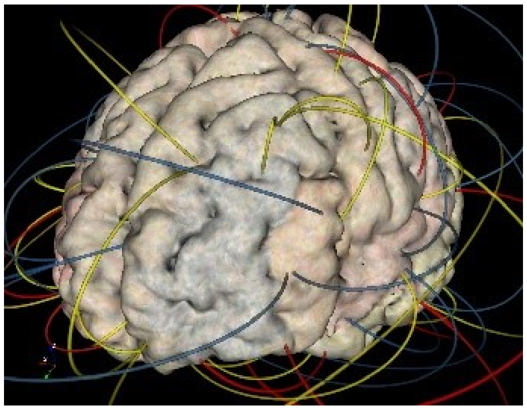
Traditional volume rendering projects voxels to a 2D screen defined in a specific viewing direction. Each new viewing direction will require a new rendering. Therefore, users need to continuously rotate and transform the volumetric object to generate different views, but never have the complete view in one image. Spherical volume rendering employs a spherical camera with a spherical screen. Thus, the projection process only happens once, providing a complete image from all angles.

Spherical ray casting

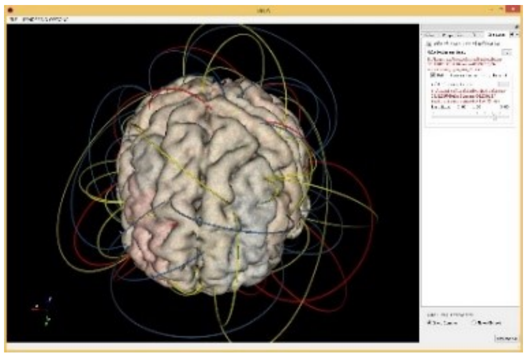
A spherical ray casting approach is taken to produce a rendering image on a spherical surface. A map projection will then be applied to construct a planar image (volume map). The algorithm includes three main steps:

(a) ROIs with noise textures (perspective 1)

(b) ROIs with noise textures (perspective 2)



(c) ROIs with noise textures and color bended edges (perspective 1)



(d) ROIs with noise textures and color bended edges (perspective 2)

Fig. 4.7. Examples of connectome networks with noise patterns.

1. Define a globe as a sphere containing the volume. The center and radius of the sphere may be predefined or adjusted interactively.
2. Apply spherical ray casting to produce an image on the globes spherical surface (ray casting algorithm).

3. Apply a map projection to unwrap the spherical surface onto a planar image (similar to the world map).

Rays are casted towards the center of the global from each latitude-longitude grid point on the sphere surface. In brain applications, the center of the global needs to be carefully defined so that the resulting image preserves proper symmetry, as shown in Figure 4.8.

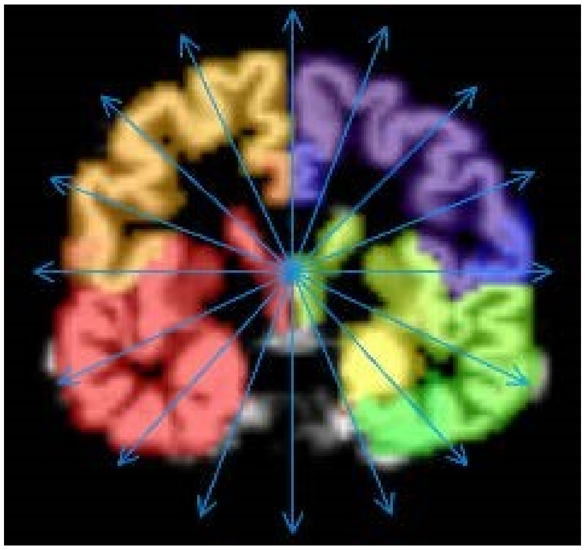


Fig. 4.8. Ray casting towards the center of the brain (sliced).

Along each ray, the sampling, shading and blending process is very similar to the regular ray casting algorithm [33, 36]. The image produced by this ray casting process on the spherical surface will be mapped to a planar image using a map projection transformation, which projects each latitudelongitude grid point on the spherical surface into a location on a planar image. There are many types of map projections, each preserving some properties while tolerating some distortions. For our application, we choose to use Hammer-Aitoff Projection, which preserves areas but not angles. Details of this map projection can be found in [108].

Layered Rendering

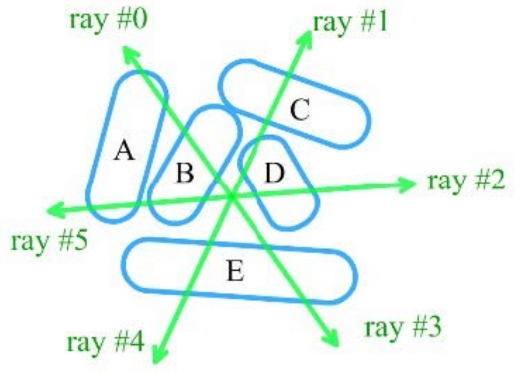
Volume rendering often cannot clearly show the deep interior structures. One remedy is to use layered rendering. When objects within the volume are labelled (e.g. segmented brain regions), we can first sort the objects in the spherical viewing direction (i.e. along the radius of the sphere), and then render one layer at a time. The spherical viewing order can usually be established by the ray casting process itself as the rays travel through the first layer of objects first, and then the second layer, etc. If we record the orders in which rays travel through these objects, we can construct a directed graph based on their occlusion relationships, as shown in Figure 4.9. Applying a topological sorting on the nodes of this graph will lead to the correct viewing order.

Since the shapes of these labelled objects may not be regular or even convex, the occlusion orders recorded by the rays may contradict each other (e.g. cyclic occlusions). Our solution is to define the weight of each directed edge as the number of rays that recorded this occlusion relationship. During the topologic sorting, the node with minimum combined incoming edge weight will be picked each time. This way, incorrect occlusion relationship will be kept to the minimum.

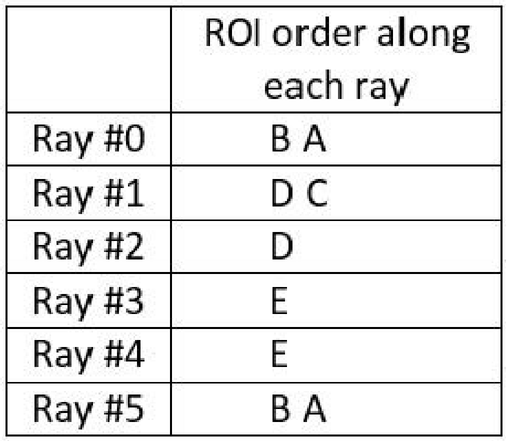
Using a spherical volume rendering algorithm, we can generate a 2D brain map that contains all the ROIs in one image. This allows the users to view clearly relationships between different ROIs and the global distributions of network attributes and measurements for feature selection and comparison.

Figure 4.10(a) shows a brain map generated by SVR without any ROI labelling. Figure 4.10(b) shows the same brain map with color coded ROI labels.

Layered rendering was also applied to brain ROIs. With opacity at 1, Figure 4.10 shows the first layer of the ROIs. Figure 4.11 shows all the layers. Different scaling factors are applied to the layers to adjust their relative sizes. This is necessary because the spherical ray casting will create enlarged internal ROIs, just like perspective projection will make closer objects larger, except that in this case the order is reversed.

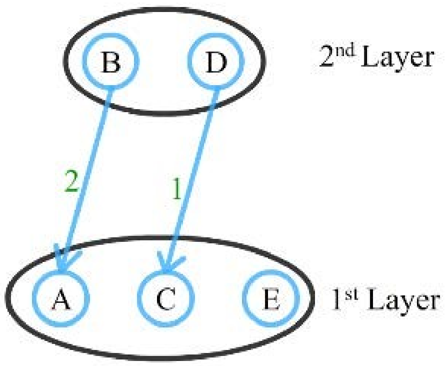


(a) 6 Rays and 5 ROIs



	ROI order along each ray
Ray #0	B A
Ray #1	D C
Ray #2	D
Ray #3	E
Ray #4	E
Ray #5	B A

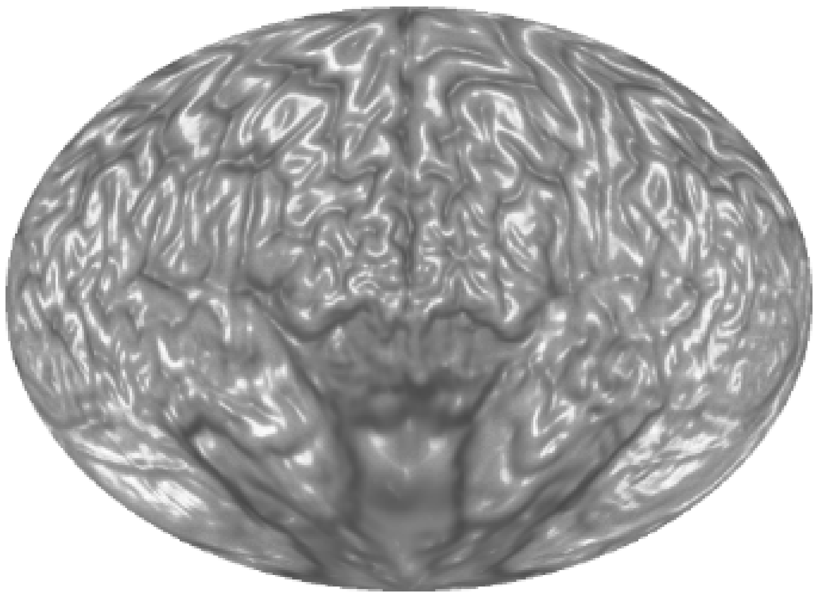
(b) Occlusion information of each ray.



(c) Occlusion graph and layers.

Fig. 4.9. An example of layer sorting for regions of interest (ROIs).

In the following two subsections, we demonstrate two approaches to overlay additional information on top of the brain map: (1) Encoding attribute information onto a texture image and then mapping the texture to the ROI surface; (2) Drawing network edges directly over the brain map. Below, we apply the first approach to an application of visualizing discriminative patterns among multiple classes. In addition, we combine both approaches to visualize fMRI data and the corresponding functional connectivity network.


(a) Without ROI labels

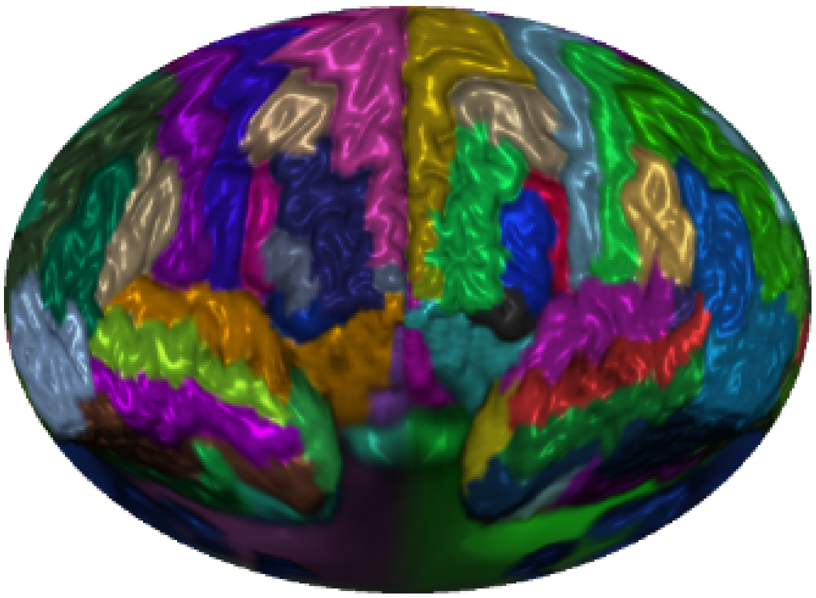
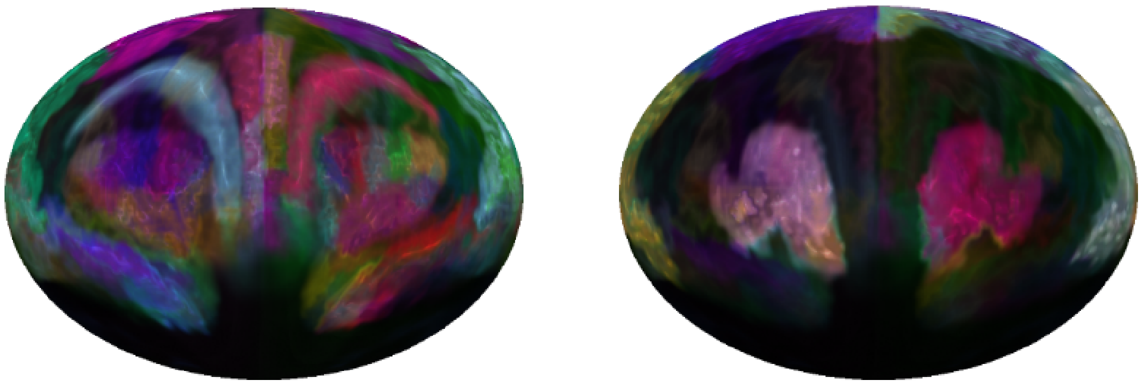
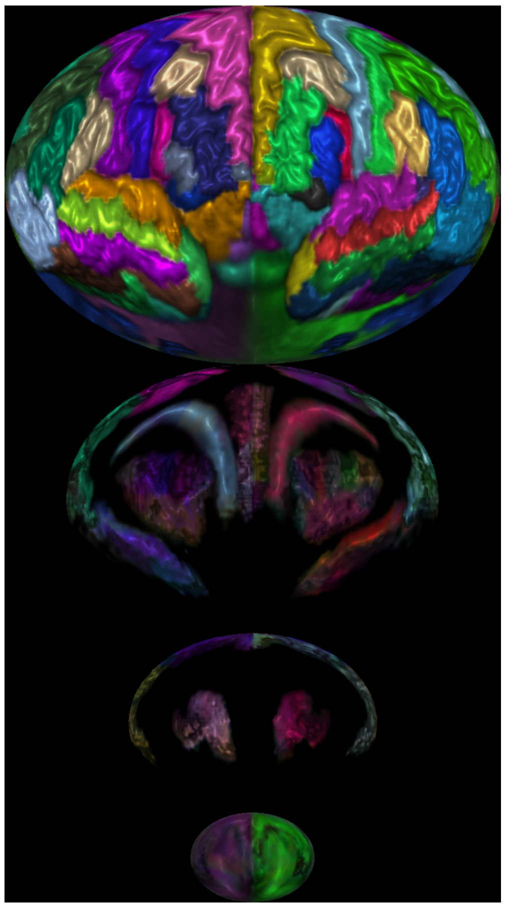

(b) With ROI labels.

Fig. 4.10. A brain map.



(a) 2nd layer.

(b) 3rd layer.



(c) all layers stacked.

Fig. 4.11. Layers of a brain map.

4.1.5 Visualizing fMRI Data and Discriminative Pattern

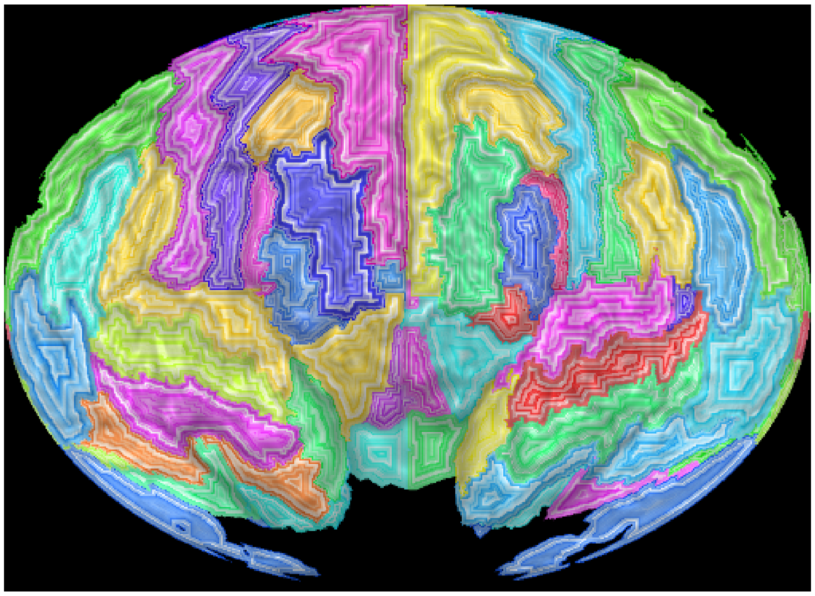
Figure 4.12 shows the fMRI textured brain map for the first two layers. Figure 4.14 shows the network edges across multiple layers for both time-series and multi-disease textures.

4.1.6 User Interface and Interaction

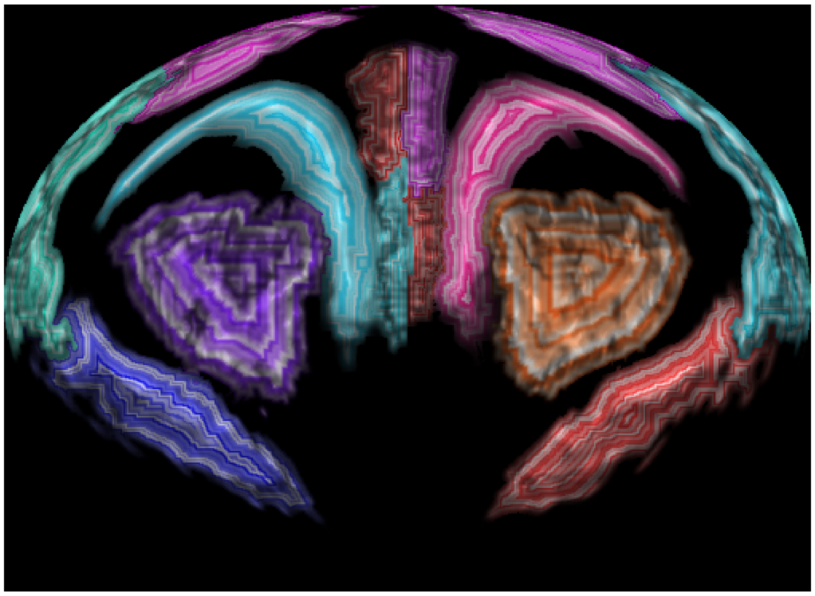
Compared with traditional volume rendering in the native 3D space, this approach views the brain from its center. On one hand, this can reduce the volume depth it sees through. On the other hand, it renders ROIs in a polar fashion and arranges ROIs more effectively in a bigger space. With more space available, it is easier to map attributes onto the ROIs and plot the brain networks among ROIs. Compared with traditional 2D image slice view, this approach can render the entire brain using much fewer layers. The user interface (Figure 4.15) is flexible enough for users to adjust camera locations and viewing direction. Users can conveniently place the camera into an ideal location to get an optimized view. Users can also easily navigate not only inside but also outside the brain volume to focus on the structures of their interest or view the brain from a unique angle of their interest.

4.1.7 Performance and Evaluation

The SVR algorithm is implemented on GPU with OpenCL [32] on NVIDIA GeForce GTX 970 graphics card with 4GB memory. We pass the volume data to kernel function as image3d_t objects in OpenCL in order to make use of the hardware-accelerated bilinear interpolation when sampling along each ray. The normal of each voxel, which is required in BlinnPhong shading model, is pre-calculated on CPU when the MRI volume is loaded. The normal is also treated as a color image3d objects in OpenCL, which can save lots on time on interpolation. We make each ray one OpenCL work-item in order to render each pixel in parallel. The global work-item



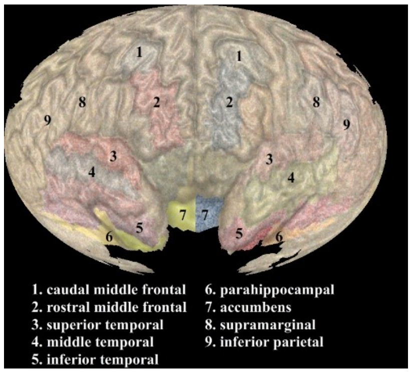
(a) 1st layer textures



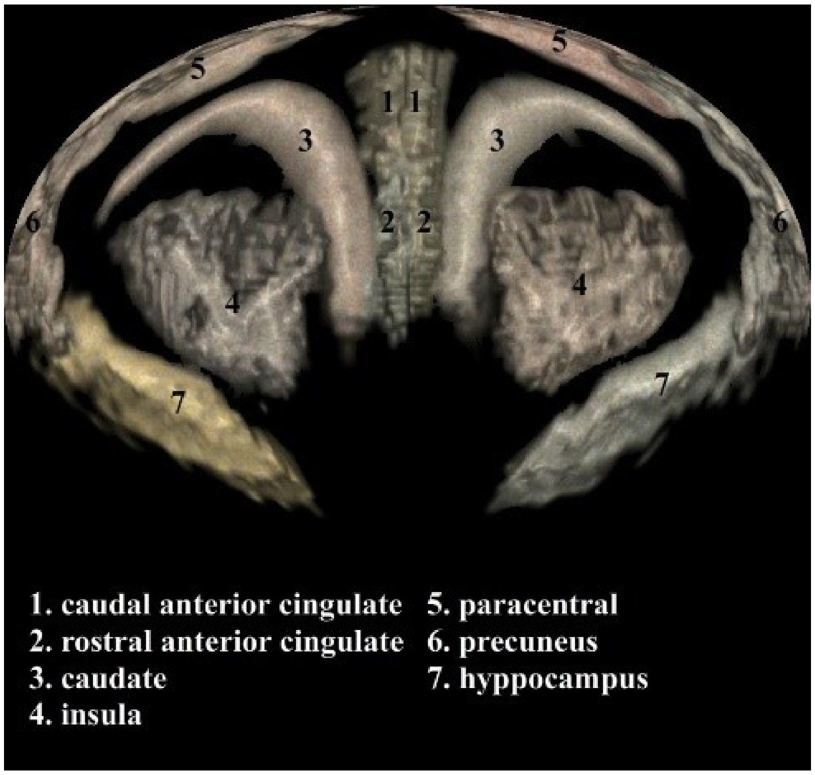
(b) 2nd layer textures.

Fig. 4.12. Textured brain map for fMRI data.

size is the size of the viewport. The performance depends on the output image size,

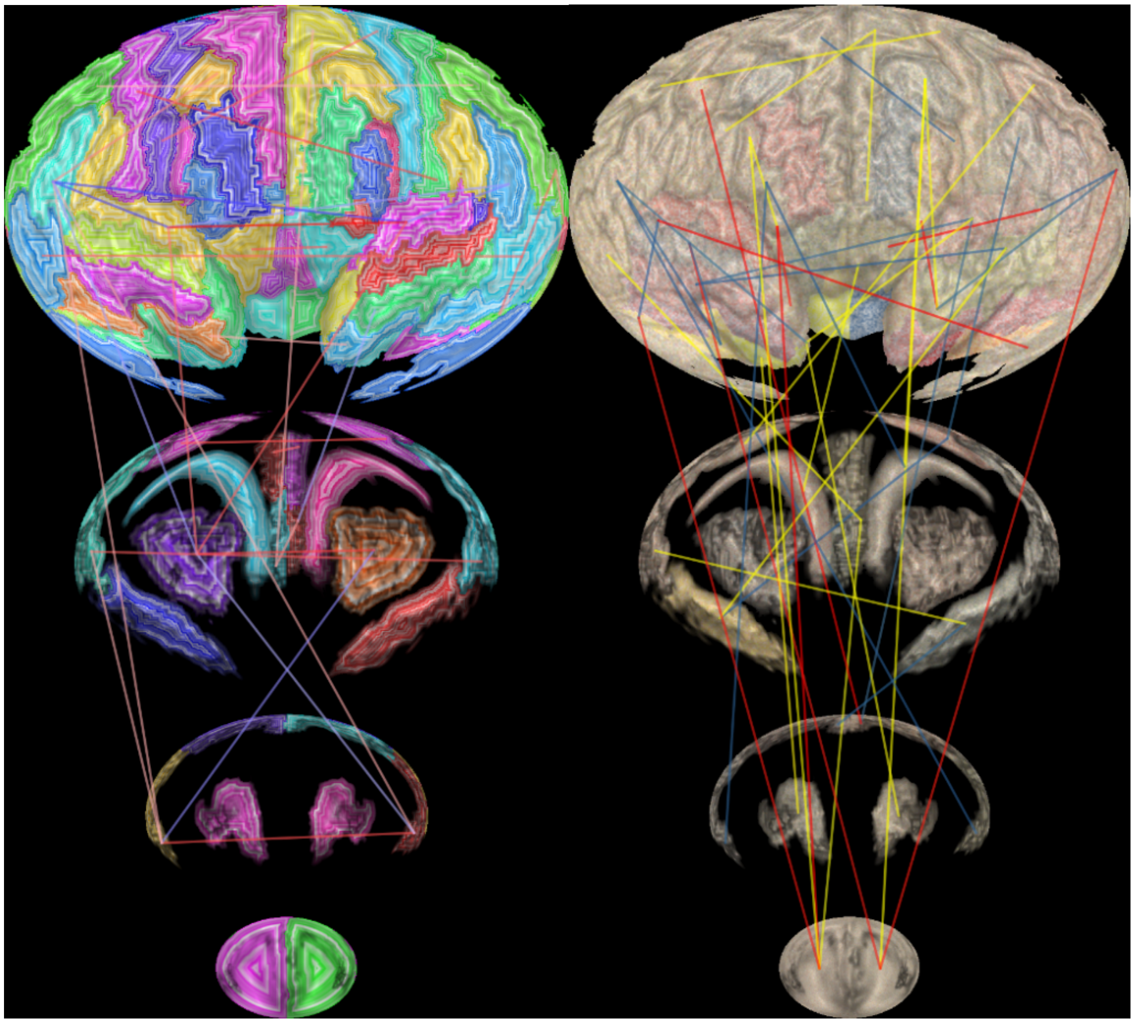


(a) 1st layer textures.



(b) 2nd layer textures. A noise pattern is applied with 3 colors representing the 3 categories (i.e., red for HC, yellow for MCI, and blue for AD).

Fig. 4.13. Textured brain map for disease classification.



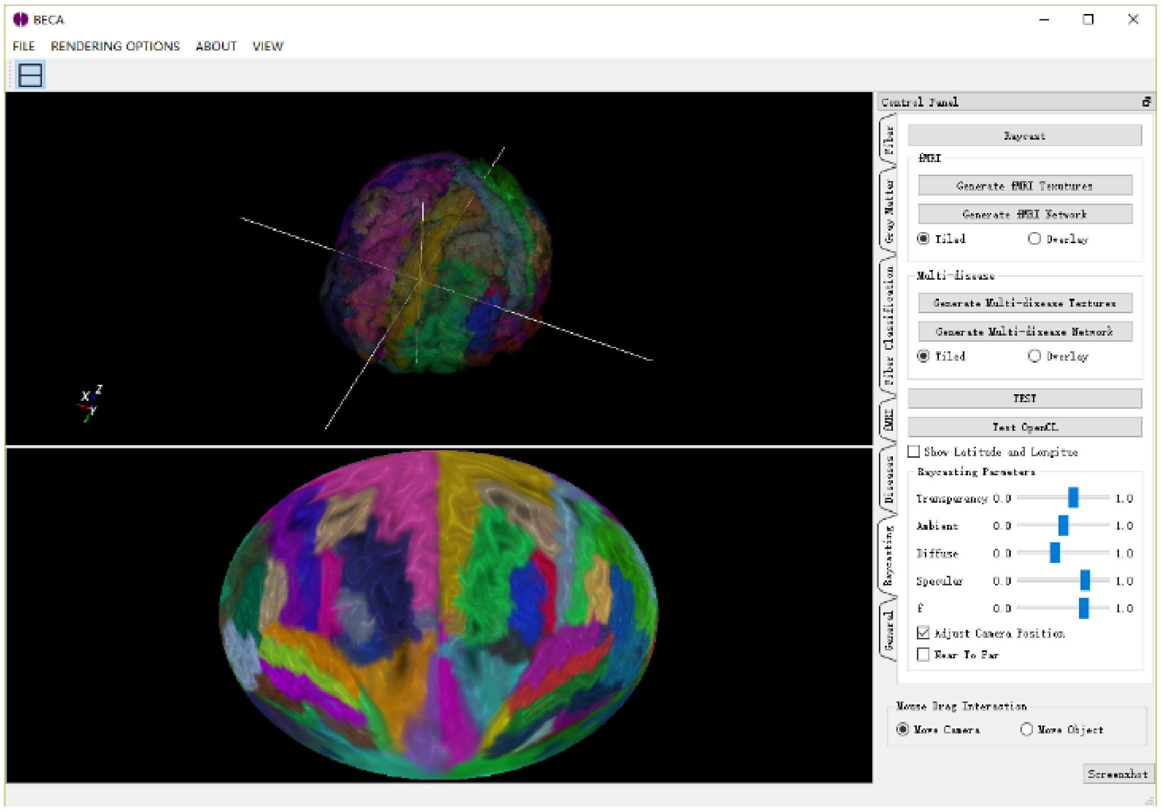
(a) Network edges over multiple layers for time-series textures.

(b) Network edges over multiple layers for multi-disease textures.

Fig. 4.14. Displaying network edges between layers.

which is shown in Table 4.1. With an 800 × 600 viewport size, the performance is around 29.41 frames per second.

We have developed tools using Qt framework and VTK to allow user to interact with the 2D map [33, 34]. Users can drag the sphere camera around in the 3D view and the 2D map will update in real-time. A screenshot of the user interface is shown in Figure 4.15. The upper half is the brain in 3D perspective view while the lower half



[✓ Tiled] Overlay

[✓ Tiled] Overlay

☐ Show Latitude and Longitue

Transparency	0.0		1.0
Ambient	0.0		1.0
Diffuse	0.0		1.0
Specular	0.0		1.0
f	0.0		1.0

☑ Adjust Camera Position
☐ Near To Far

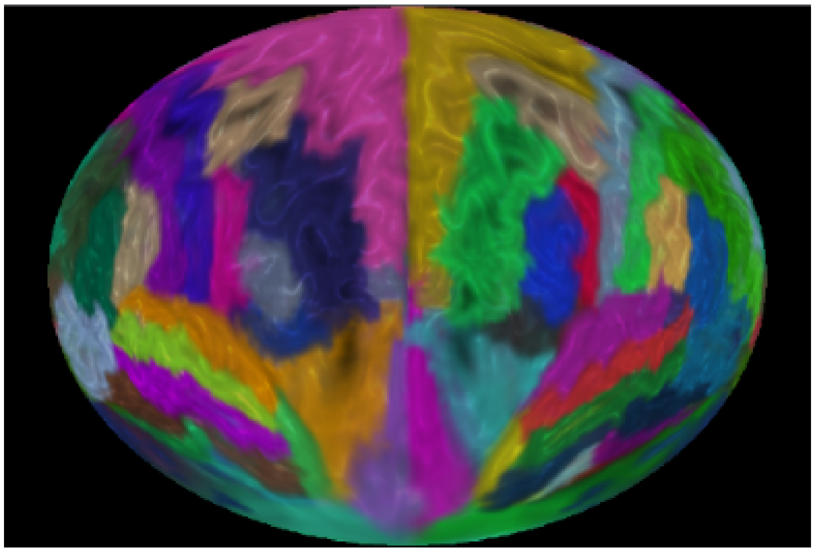
[✓ Move Camera] Move Object

Fig. 4.15. A screenshot of the user interface. When user drag the camera (intersection of the white lines) on the top, the 2D map on the bottom which will be re-rendered in real-time.

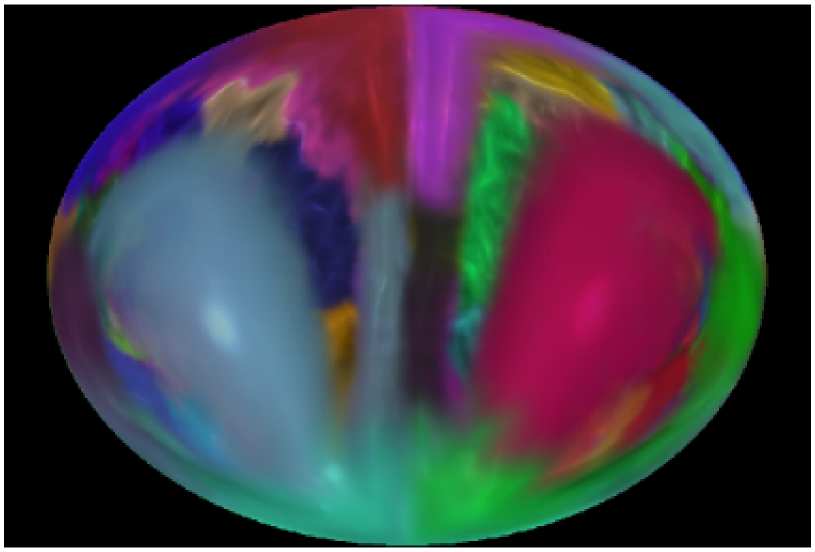
Table 4.1.
Frame rates for different output resolutions on NVIDIA GTX 970

Output Resolution	Avg. fps
640 × 480	45.45
800 × 600	29.41
1024 × 768	11.76
1600 × 1200	7.04

is the 2D brain map generated by the SVR algorithm. When user move the position of spherical camera (intersection of the white lines in Figure 4.15) in the 3D view, the 2D map will change accordingly. The software enable user to navigate in the 3D brain and build the visual correspondence between the 3D and 2D representation. We also provide users with a switch to reverse the direction of rays. As shown in Figure 4.16(a), rays are travels outward and we can see the exterior of the brain. On the contrary, when we reverse the direction of the ray in Figure 4.16(b), we can see the interior structures of the brain. We demonstrated our prototype system and the resulting visualization to the domain experts in Indiana University Center for Neuroimaging. The following is a summary of their evaluation comments.



(a) Rays travel outwards



(b) Rays travel inwards.

Fig. 4.16. Reverse the direction of rays.

Evaluation on the visualization of the discriminative pattern. The discriminative pattern shown in Figure 4.13 has the promise to guide further detailed analysis for identifying disease-relevant network biomarkers. For example, in a recent Nature Review Neuroscience paper [35], C. Stam reviewed modern network science findings in neurological disorders including Alzheimers disease. The most consistent pattern the author identified is the disruption of hub nodes in the temporal, parietal and frontal regions. In Figure 4.13, red regions in superior temporal gyri and inferior temporal gyri indicate that these regions have higher connectivity in HC than MCI and AD. This is in accordance with the findings reported in [35]. In addition, in Figure 4.13, the left rostral middle frontal gyrus shows higher connectivity in HC (i.e., red color) while the right rostral middle frontal gyrus shows higher connectivity in AD (i.e., blue color). This also matches the pattern shown in the Figure 4.3 of [35], where the hubs at left middle frontal gyrus (MFG) were reported in controls and those at right MFG were reported in AD patients. These encouraging observations demonstrate that our visual discriminative patterns have the potential to guide subsequent analyses.

Evaluation on the visualization of fMRI data and functional network. It is helpful to see all the fMRI signals on the entire brain in a single 2D image (Figure 4.14). Drawing a functional network directly on the flattened spherical volume rendering image (Figure 4.14) offers an alternative and effective strategy to present the brain networks. Compared with traditional approach of direct rendering in the 3D brain space, while still maintaining an intuitive anatomically meaningful spatial arrangement, this new approach has more spatial room to work with to render an attractive network visualization on the background of interpretable brain anatomy. The network plot on a multi-layer visualization (Figure 4.14) renders the brain connectivity data more clearly and effectively.

Evaluation on the user interface and interaction. Compared with traditional volume rendering in the native 3D space, this approach views the brain from its center. On one hand, this can reduce the volume depth it sees through. On the other hand, it renders ROIs in a polar fashion and arranges ROIs more effectively in a bigger space. With more space available, it is easier to map attributes onto the ROIs and plot the brain networks among ROIs. Compared with traditional 2D image slice view, this approach can render the entire brain using much fewer layers (4 in our case) than the number of image slices (e.g., 256 slices in a conformed 1 mm3 isotropic brain volume).

The user interface (Figure 4.15) is flexible enough for users to adjust camera locations and viewing direction. Users can conveniently place the camera into an ideal location to get an optimized view. Users can also easily navigate not only inside but also outside the brain volume to focus on the structures of their interest or view the brain from a unique angle of their interest.

4.2 Interactive Machine Learning

In this section, I will introduce the second part of my study: a framework of interactive machine learning by visualization, and the application of this framework to two test examples.

4.2.1 System Overview

Our goal is to develop a new interactive and iterative learning technique built on top of any machine learning algorithm so that the user can interact with the machine learning model dynamically to provide feedback to incrementally and iteratively improve the performance of the model. Although there can be many different forms of user feedback, such as knowledge input, features selection and parameters setting, in this paper we focus primarily on adding the optimal subset of training data samples such that the added training samples can provide maximal improvement of the model using minimum number of additional training points. Hence, the problem statement can be formulated as follows:

Let F be the feature space of a machine learning algorithm, $X = \{x_1, x_2, ..., x_n\} \subset F$ be the starting training set, and $Y = \{y_1, y_2, ..., y_m\} \subset F$ be an internal test set. We define U as user space of the same dataset containing some user defined attributes. These user defined attributes are selected based on two criteria: (1) they are part of the attributes of the original dataset; and (2) they can be used to identify data points (to be collected) easily. Let $Z = \{z_1, z_2, ..., z_m\} \subset U$ be set Y represented in the user space U, and $M_0(y) : F \rightarrow C$ be the learned model using the initial training set X, where C is the application value range (e.g. class labels or regression function values).

We want to find a set of k new data points (where k is a constant), $X' \subset F$, such that points in X' satisfy a set of user defined conditions of attribute values in U. These conditions in U is defined interactively from the visualization of the model and its test results on set Z in the user space U. The users goal is to provide additional training samples such that the learned model $M_1(y)$ using training set $X \cup X' \subset F$ is an improved model over $M_0(y)$.

The above process can continue iteratively until the performance of the model is satisfactory or until the model can no longer be improved.

This framework can be summarized by the structural flowchart in Figure 4.17. At each iteration, a machine learning model is constructed using the current training set. The model will be tested on an internal test set. The visualization engine will then visualize the model along with the labeled internal test results. Based on this visualization, the user can decide to add new samples in the areas where the model performed poorly. These new samples will be added to the current training set to enter the next iteration.

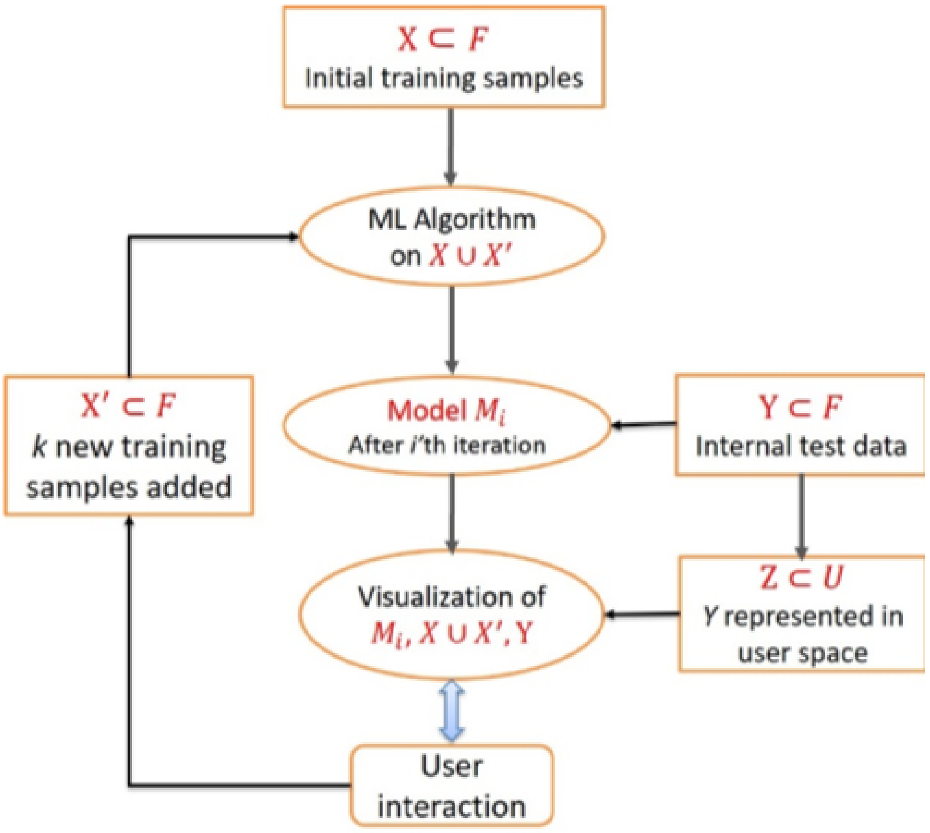


Fig. 4.17. A structural flowchart of the interactive machine learning system.

4.2.2 User Space and User Interactions

A critical idea in our interactive machine learning framework is the separation of feature space and user space. During each iteration of the learning process, the additional training data is often not readily available, and needs to be acquired separately using some easy to use attributes.

A machine learning algorithm learns a model based on the features of the training samples. These features are either pre-computed by some dimension reduction methods (e.g. PCA) or selected through some feature selection algorithms. It is generally not feasible to obtain features of any data item before the data is collected. This is particularly true for complex datasets where the collection of each data item requires significant effort and cost. For example, in medical analysis, the collection of detailed medical and health data for each patient or a control individual is very expensive and time-consuming.

In our approach, a specific subset of conditions for data is identified through the interactive visualization process and targeted for collection. Thus, the attribute conditions for this subset of data need to be something that are easy to be used for the identification and collection of data. For this reason, we define user space as a data representation space containing attributes that can be used as the identifiers of the target data subset for iterative data collection. This also means that the interactive visualization also needs to be presented in this user space so that the user can interactively define the attribute conditions for additional data samples.

A user space is typically defined by the user based on the application needs. The attributes in the user space may contain:

- **Common attributes.** These include simple common characteristics of data that can be used to identify the data easily. For instance, in medical diagnosis applications, these may include common demographic information and behavior data such as age, gender, race, height, weight, social behavior, smoking habit, etc.

- **Special attributes.** These are attributes the analysts have special interests in. For example, in bioinformatics, certain group of genes or proteins may be of special interests to a particular research problem, and can be extracted from a large database.

- **Visual attributes.** Visual data such as images or shape data maybe directly visualized as part of the user space so that the user can visually identify similar shapes or images as new samples.

Through the visualization of the model and the associated labels of the testing samples, the user can specify conditions for user space attributes to identify new training samples. This is done based on several different principles:

- **Model Smoothness.** The visualization will show the shape of the learned model at each iteration. Visual inspection of the shape of the model can reveal potential problem areas. For example, if the model is mostly smooth but is very fragmented in a certain region, it is possible that the learning process does not have sufficient data in that region.

- **Testing Errors.** Errors from the test samples can provide hints about areas where the model performs poorly. These may include misclassified samples and regression function errors. In areas with significant errors, new samples may be necessary to correct the model.

- **Data Distribution.** There may be a lack of training data in some area in the user space. This can affect the models accuracy and reliability. For example, a medical data analysis problem may lack sufficient data from older Asian female patients. To show this type of potential issues, the visualization system will need to draw not only the test samples, but also the training samples within the user space.

4.2.3 The Visualization Platform

The visualization platform in our interactive machine learning framework serves as the user interface to support user interaction and the visualization of data and the model.

Although there are many different visualization techniques for multi-dimensional data [27], we choose scatterplot matrix as our main visualization tool as it provides the best interaction support and flexibility. We also choose to use heatmap images to visualize the machine learning model within the scatterplots since it treats the machine learning model as a black box function and thus allows the approach to be machine learning algorithm independent.

Figure 4.18 shows a general configuration of the scatterplot visualization interface. The upper-right half of the matrix shows the feature space scatterplots, the lower-left half shows the user space scatterplots and the diagonal shows the errors of the corresponding feature space dimensions. Within each scatterplot sub-window, two types of visualizations will be displayed: (1) the data (training or testing data points); (2) the current learned model. Each of the 2D sub-windows can also be enlarged for detailed viewing and interaction. In principle, the dimensionality of the feature space and the dimensionality of the user space are not necessarily the same. But for convenience, we may select the same number of features and user space attributes to visualize in this scatterplot matrix. It is certainly not hard to use different numbers of variables in these two spaces.

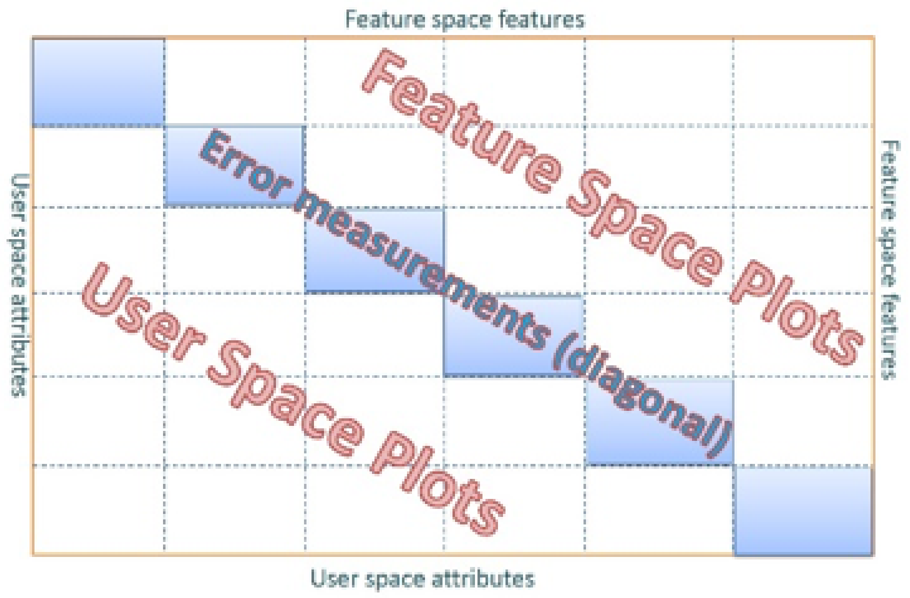


Fig. 4.18. A configuration of the scatterplot matrix visualization.

The primary challenge in this visualization strategy is the visualization of a machine learning model in a 2D subspace of the feature space or user space. A heatmap image filling approach will be used to visualize the model. Each pixel of the 2D sub-window will be sampled against the model function, and the result will be color-coded to generate a heatmap-like image. An example is shown in Figure 4.19 for a 3-class classification model.

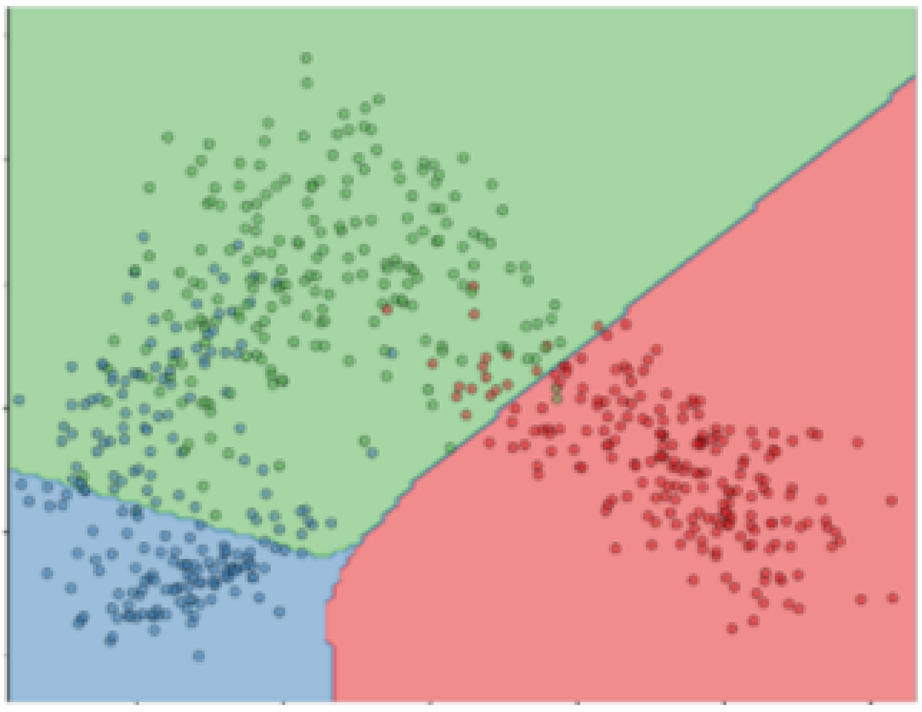


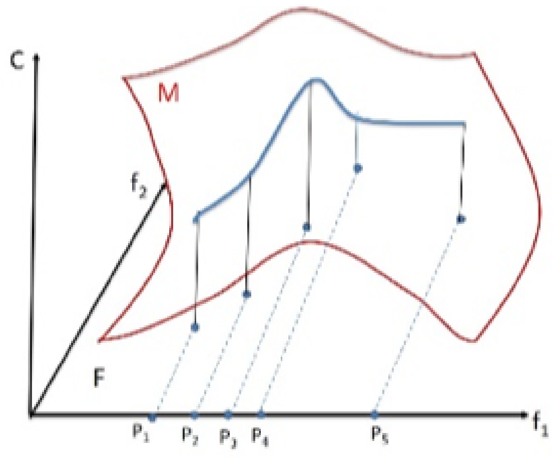
Fig. 4.19. Model visualization example.

Let the machine learning model be a function over the feature space, $M(y) : F \rightarrow C$, where F is the feature space and C is the range of the model function. The projection of the model in a 2D subspace is, however, not well defined, and hard to visualize and understand. A better way to understand and visualize the model in a 2D subspace is to draw a cross-section surface (over the 2D subspace) of the model function that passes through all training points. Mathematically, this is equivalent to the following:

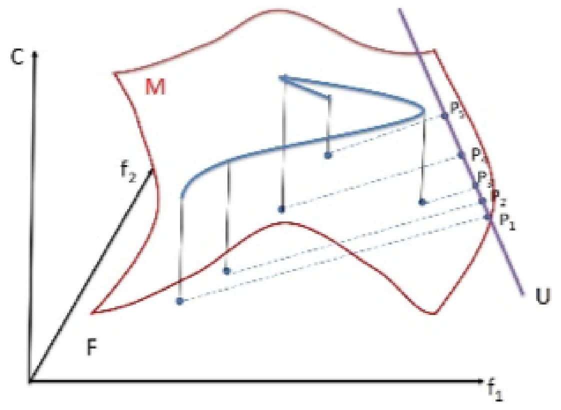
For a pixel point $P = (a, b)$ in a 2D subspace where a and b are either two feature values or two user space attributes, compute $M(y)$, where the feature vector yF at P is calculated by interpolating the feature vectors of the training samples on this 2D subspace.

Any 2D scattered data interpolation algorithm can be used here to interpolate the feature vectors. In our implementation, since we need to interpolate all pixels in a 2D sub-window, a triangulation-based interpolation method is more efficient as the triangulated interpolants only need to be constructed once. The training data samples are triangulated by Delaunay triangulation first. A piecewise smooth cubic Bezier spline interpolant is constructed over the triangulation using a Clough-Tocher scheme [28]. An alternative method is to apply piecewise linear interpolation over the triangular mesh. But the cubic interpolation provides better smoothness. Please note that this interpolation scheme interpolates only the feature vectors, which will then be inputted to the model function to generate model output values for color coding.

Figure 4.20 shows two types of cross-sections. For simplicity of illustration, we use a 1D analog to the 2D cross-sections. So, the sample points on a 1D axis in the figure should be understood as the sampling points on a 2D scatterplot sub-window. Here $(f1, f2)$ is the feature space. C is the model value, U is a 1D subspace of the user space. P1 to P5 are the training samples we use for interpolation. In Figure 4.20(a), f1 axis is a subspace we use to visualize the model in the feature space. In Figure 4.20(b), U is a subspace we use to visualize the model in the user space. In this figure, we assume U is a linear combination (rotation) of the feature dimensions. But U sometimes can be wholly or partially independent of the features. In that case, interpolation will just simply be done within the user space similarly as in Figure 4.20(a). Since we have many 2D sub-windows in the scatterplot matrix, the combinations of these cross-sections provide a cumulative visual display of the model function at every iteration of the learning process.



(a)



(b)

Fig. 4.20. 1D illustrations of model visualization as cross-sections in feature space and in user space.

4.2.4 Test Applications

The framework described in the last section has been implemented using Python and a Python 2D plotting library: Matplotlib. In this section, we will apply this framework to two different types of test applications: handwriting recognition (classification) and human cognitive score prediction (regression) using real world datasets.

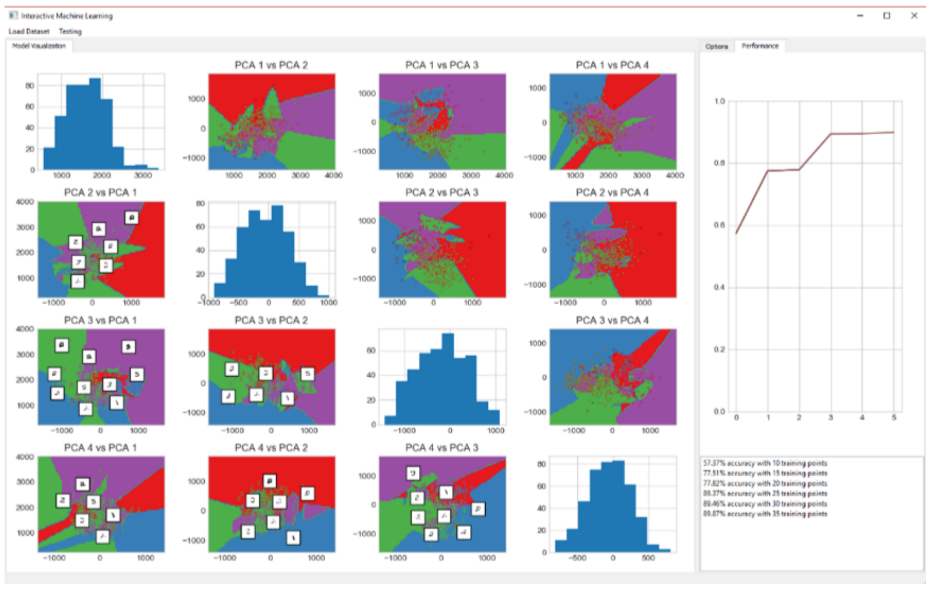
Handwriting Recognition

In this case study, we apply our interactive machine learning approach to the classification (recognition) of handwriting digits. The well-known publicly available MNIST handwritten digits dataset from the MNIST database [29] is used. For better illustration effect, we will narrow the recognition scope to four classes of digits: 0s, 1s, 2s, and 3s. For these 4 classes, we have 24673 training points and 4159 testing points. We also picked 123 points (0.5%) out of the training set for internal test to guide the interactive process.

Each original data point contains a fixed sized 2D image. Principal Component Analysis (PCA) is applied to the pixel arrays of these images. The top 10 principal components are used as feature vector in a Support Vector Machines classification algorithm. This SVM process is very similar to a typical face recognition system [30].

In this application, the feature space is the principal component space. If we only use the top four features (PCs) for visualization and interaction, we will have a 4 by 4 scatterplot matrix. Since this is an image dataset, the user space contains the original pixels. It is easier to simply display small icons of some of the original images within the user space scatterplots. These icons can be enlarged when clicked by the user.

Figure 4.21 shows an interface for this interactive session with four features. The scatterplot matrix is symmetric, but the lower-left sub-windows show icons of some of the original images, which serve as a user space. Each diagonal sub-window shows a histogram of the distribution of the misclassified points in each feature dimension. It is certainly possible to use the shape information of the misclassified points to retrieve similar new samples from the large database, which can perhaps be automated. In our experiment, we focus only on interactive operations. New samples are added in areas where there are too many misclassified samples or the classification boundaries appear fragmented. The process started with only 10 training samples. In each iteration, 5 new samples are added at an area clicked by the user.



```
57.37% accuracy with 10 training points
77.51% accuracy with 15 training points
77.82% accuracy with 20 training points
89.37% accuracy with 25 training points
89.46% accuracy with 30 training points
89.87% accuracy with 35 training points
```

Fig. 4.21. Interactive machine learning interface for 4-class handwriting classification.

Figure 4.22 shows a performance chart for this experiment. The orange line represents the performance using randomly selected samples, and the blue line represents the performance using interactively selected samples. Since this problem is relatively easy, the performance curve converges quickly after about 250 points. But the blue line reaches the near-peak performance much earlier at about 75 points. Figure 4.23 shows a sequence of the interactions that led to iterative model improvement.

Human Cognitive Score Prediction

Understanding the structural basis of human cognition is a fundamental problem in brain science. Many studies have been performed to predict the cognitive outcomes from measures captured by Magnetic Resonance Imaging (MRI) scans of the brain [31-34]. In this case study, we apply our interactive machine learning approach to the prediction of cognitive performance using MRI data coupled with relevant demographics and behavior information.

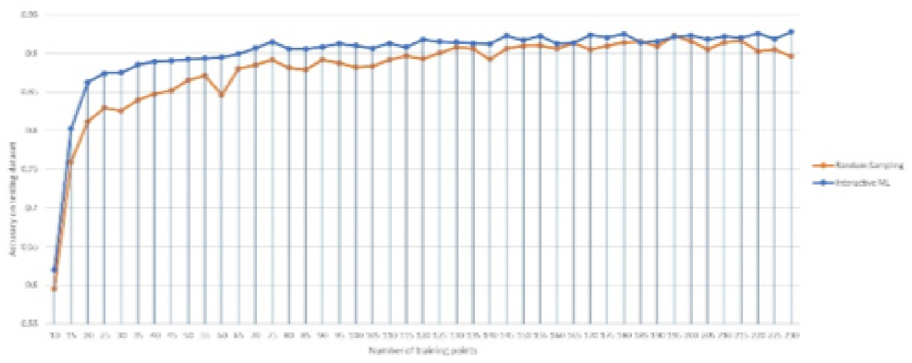


Fig. 4.22. Performance chart for handwriting classification experiment.

The data studied in this work were downloaded from the Human Connectome Project (HCP) database [35-37]. HCP is a major NIH-sponsored endeavor that has acquired and published brain connectivity data plus other neuroimaging, behavioral, and genetic data from 1200 healthy young adults. Its goal is to build human brain network map (i.e., connectome) to better understand the anatomical and functional connectivity in relation to cognitive and behavior outcomes within the healthy human brain.

There are four types of attributes from the HCP database: (1) the MiniMental State Examination (MMSE) score, which is the cognitive outcome studied in this work; (2) structural MRI measures, including volume measures and cortical thickness measures of regions of interests generated by the FreeSurfer software tool [38]; (3) demographical measures such as age, race, weight, height, BMI, etc.; and (4) behavioral measures. Our computational task is to predict the MMSE score using the MRI, demographical and behavior measures.

A total of 1177 subjects with complete cognitive, imaging, demographical, and behavior information were included in our study. 589 subjects are used as test set and 588 subjects are used as training set. We selected 5% (about 30 samples) of this training set as internal test set to guide the user interaction. A principal component analysis (PCA) is used for feature selection. A support vector regression (SVR)

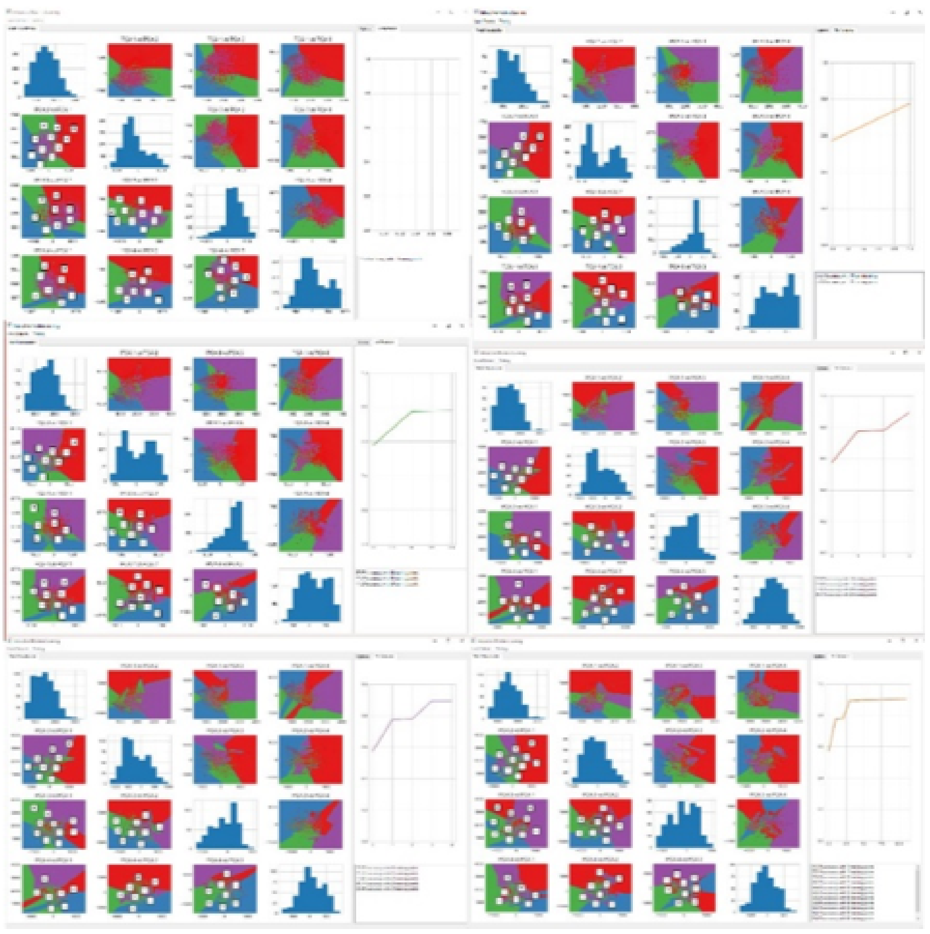
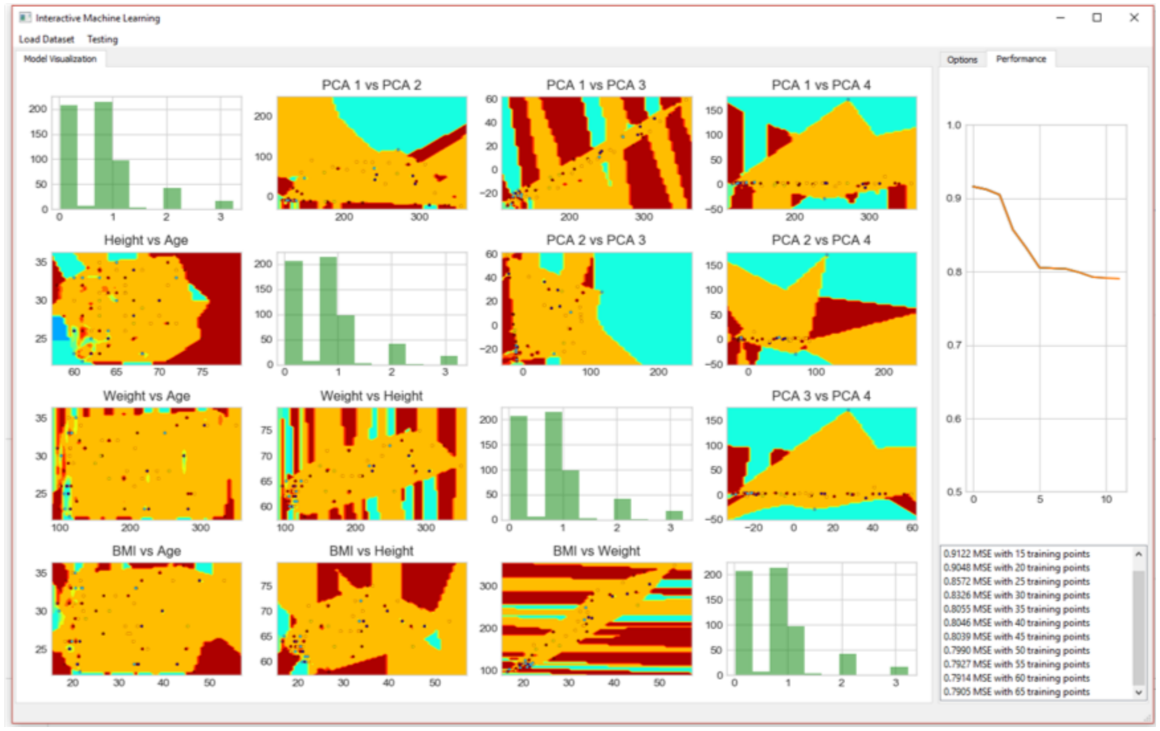


Fig. 4.23. A sequence of model improvement iterations.

technique is applied to the top 10 principal components (PCs) to obtain a regression model in each iteration for the prediction the MMSE scores [39, 40]. The predicted MMSE scores are then color coded to generate the heatmap images in the 2D sub-windows. The iteration starts with 10 initial training samples. In each iteration, 5 new samples are added at an area clicked by the user.

Figure 4.24 shows a screen shot of the interface. The top four principal components are used as the features in the scatterplots. The user space attributes used in this visualization include the patients weight, height, age, and body mass index (BMI). The diagonal sub-windows show the histograms of the distributions of the regression errors for the individual feature dimensions. We again only focus on adding new samples in areas where there are too many mismatches of colors between the model and the samples or the regression heatmap appears too fragmented. In practice, experts may also use other professional knowledge to add samples that relate to a particular hypothesis. From Figure 4.24, we can see that the scores are very flat in most of the regions but can change quite drastically within some isolated small region.



```
0.9122 MSE with 15 training points
0.9048 MSE with 20 training points
0.8572 MSE with 25 training points
0.8326 MSE with 30 training points
0.8055 MSE with 35 training points
0.8046 MSE with 40 training points
0.8039 MSE with 45 training points
0.7990 MSE with 50 training points
0.7927 MSE with 55 training points
0.7914 MSE with 60 training points
0.7905 MSE with 65 training points
```

Fig. 4.24. Interactive machine learning interface for human brain data regression.

Figure 4.25 shows the performance chart for this iteratively built model. The orange line represents the results using randomly selected samples, and the blue line represents the results using interactively selected samples. We first applied the support vector regression using the entire 588 training set. The resulting mean error is 0.8. The chart shows that the interactive model converge quickly to the optimal performance (0.8 error) after about 80 training samples while the randomly selected training samples struggle to converge. Figure 4.26 shows a sequence of interactions that led to iterative model improvement.

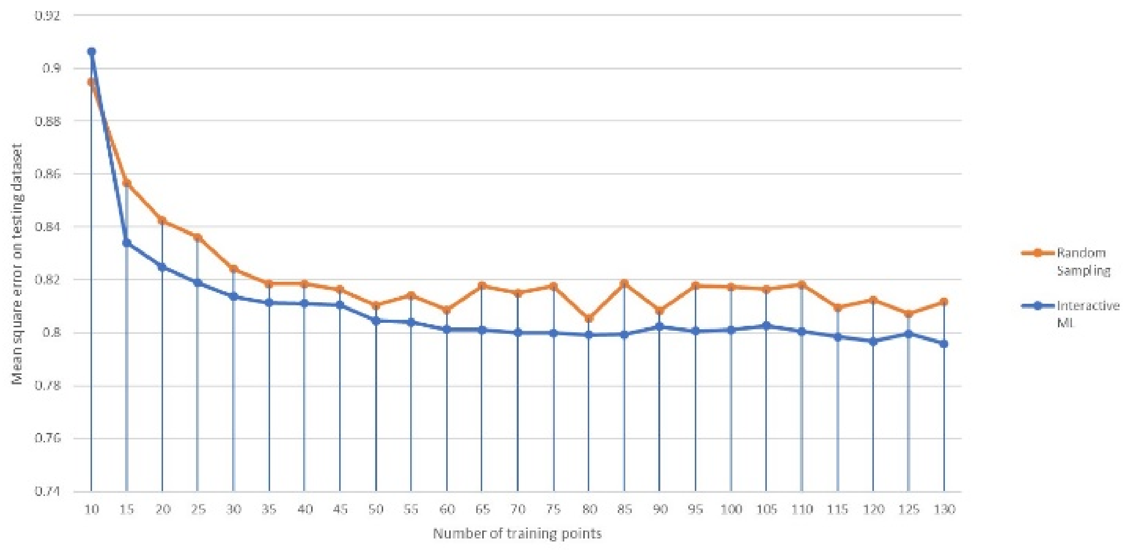


Fig. 4.25. Performance chart for brain data regression.

4.3 Interactive Visualization of Deep Learning for 3D Brain Data Analysis

Recent advances in deep neural networks and other machine learning techniques have led to unprecedented breakthroughs in many big data analysis problems and applications. The intrinsic complexity in deep learning, introduced by multiple hidden layers and massive combinations of features and weights, makes the model hard to interpret. Therefore, in most scenarios, deep learning models are used as black boxes [109]. Visualization has been playing an increasingly important role in data mining and data analytics. This transformation of visualization from data viewing to becoming an integrated part of the analysis process led to the birth of the field of visual analytics [110]. In visual analytics, carefully designed visualization processes can effectively decode the insight of the data through visual transformations and

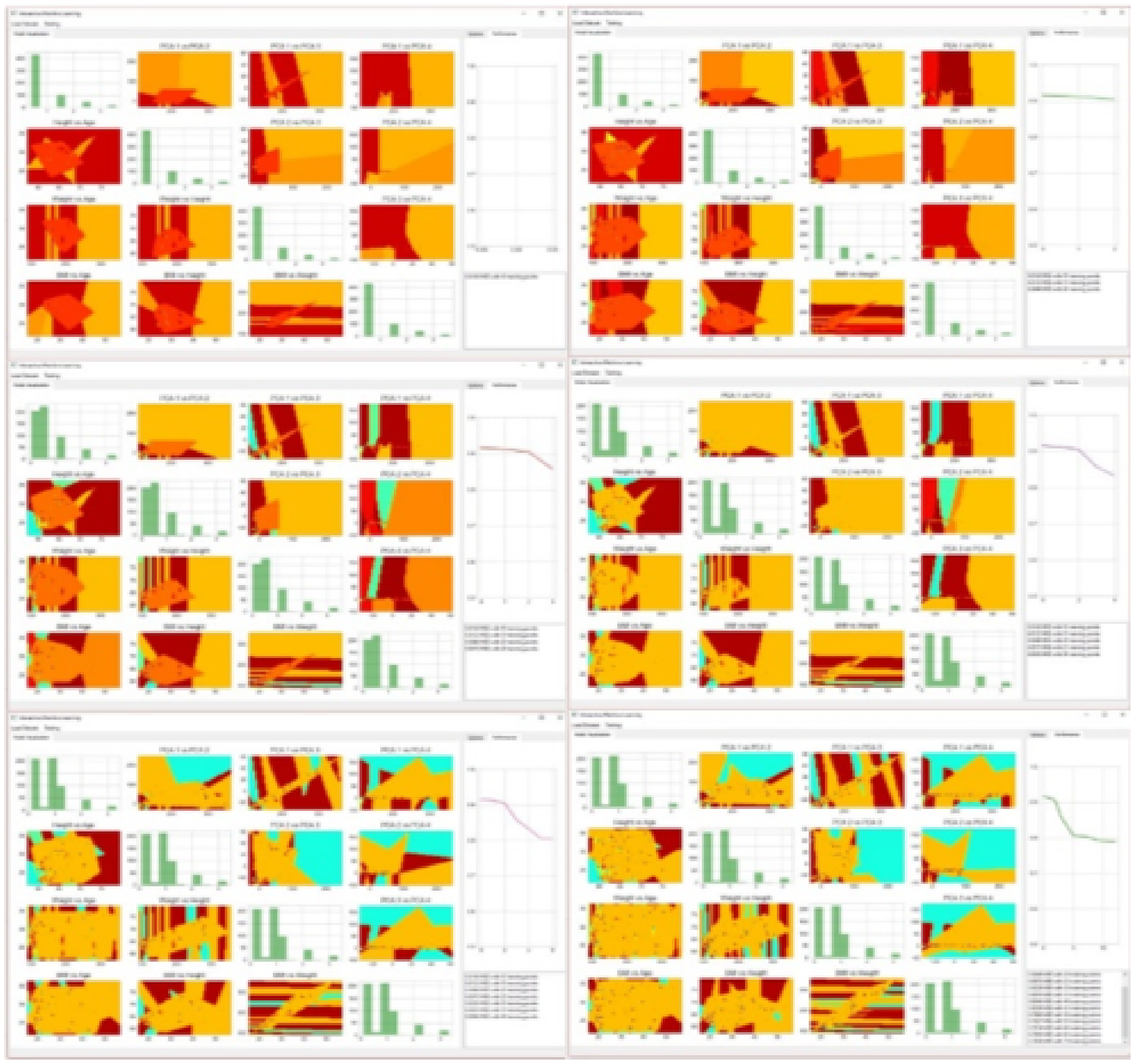


Fig. 4.26. A sequence of model improvement iterations for MMSE score prediction.

interactive exploration. Many successful applications of visual analytics have been published in recent years, ranging from bioinformatics and medicine to engineering and social sciences.

As automatic methods, machine learning algorithms, particularly deep learning algorithms, provide the users very little information about how and why the algorithms work or fail. The underlying deep learning models are also designed primarily for the convenience of learning from data, but they are not easy for the users to understand or interact with. Interactive visualization can provide an effective mechanism to help the users understand and interact with the deep learning process [111]. This has two important potential benefits.

(1) Understanding. Understanding the deep learning process in the context of the application is particularly important for biomedical applications in which how the biological processes and pathways are involved is sometimes more important than the result itself. It is also difficult to improve the efficiency and performance of the algorithms without a clear understanding of how and why the different components work in deep learning algorithms with many layers and interconnected components.

(2) Knowledge Input. In the biomedical field, human knowledge and expertise is critical in producing biologically meaningful results. Therefore, human input during the deep learning process is sometimes desirable and even necessary to generate the best and the most meaningful result.

Our goal is to develop an interactive visualization framework in a deep learning environment to provide contextual associations with the application such that the user can observe and understand the evolution and performance of the internal layers and their feature selections. Furthermore, user will also be able to interact with the system through the visualization tools to provide feedback that may alter the learning process, for example changing the weight of an edge or adding/removing a node.

In order to better understand and effectively interact with the deep learning process, the user needs to be able to associate the deep learning models intermediate layers and their nodes with application specific information to make contextual judgements. This requires the visualization system to provide not only a visual representation for the deep neural networks, but also the application specific visual information associated with the various components of the neural networks. In other words, the system needs to be able to trace from the intermediate nodes and edges back to the original input data to generate meaningful information in the context of the application. Such contextual associations can also help the user make decisions on whether there is a valid reason to alter the learning process, and if so how this can be achieved through the editing operations of the intermediate layers.

In this chapter, we propose an interactive visualization approach to the understanding of and interaction with a deep learning algorithm for 3D brain images data. The goal is to train a classifier for the diagnosis of Alzheimer's Disease (AD). Although Alzheimer's Disease is currently not curable, early diagnosis is extremely important for patients quality of life. Furthermore, interactive visualization of the deep learning process may reveal interesting and meaningful biological interpretations that could provide potential benefits for the development of a cure. Our primary contributions in this paper include:

(1) An effective interactive visualization framework that provides direct visual associations of the deep neural network structures with the anatomical structures of the brain. It also provides user editing operations to alter the internal layers to re-train the model (with a new initial condition) when necessary.

(2) A recursive backpropagation algorithm to compute the weights of the input features in each node in the hidden layers, which can lead to the anatomical or biological interpretation of the hidden layer nodes.

(3) An effective interactive brain rendering tool that combines 3D rendering and a 2D atlas rendering based on a spherical ray-casting algorithm.

4.3.1 The Visualization Framework

The overall design of our deep learning visualization system is shown in Figure 1. The system has three visual components:

(1) Network structure window. It shows the structures of the different layers of the deep neural network. All layers are drawn with their nodes and the edges connecting nodes between neighboring layers. The intensity / transparency of these edges are weighted by their edge weights. Zooming and scaling are implemented to allow the user to explore large networks. This window provides a structural overview

of the learning process. It allows user to pick any of the nodes in the hidden layers to show the corresponding brain regions and their weights, calculated by our importance backpropagation algorithm. The user can also interactively edit and manipulate nodes and edges to make changes to the model. The interaction part will be further discussed in Section 4.3.3.

(2) Rendering window. This is where the 3D anatomical structure of a brain will be rendered. The regions of interest are segmented, and can be rendered differently. For a given (picked) node, the weights of the different brain regions associated with this node will be color-coded in the rendering window. Two different types of brain image rendering will be provided. One is the regular 3D rendering of brain cortex surfaces with rotation and zooming. The weights of the brain regions will be mapped to different colors and transparencies for surface rendering. The second type is an innovative brain map as a flattened brain atlas, generated using a spherical volume rendering algorithm. This complements the 3D surface rendering in the sense that it shows all regions in one static 2D brain map to facilitate comparisons and grouping.

(3) Control window. This is a sub-window that controls various parameters for the visualization system such as intensity and transparency mapping, color mapping and thresholding.

Figure 4.27 shows the overall system architecture and flow. This chart only shows the interactive visual exploration part of the deep learning process. The formal cross-validation process is not shown in this flowchart, as cross-validation is carried out after the visual exploration stage. Thus, we can consider these visual test subjects an internal test set that is used to improve the performance of the learning model before cross-validation is applied.

4.3.2 Importance Factor Backpropagation

Once a deep learning model is trained, the user can run the model on a given test subject as input to observe how the model works with a certain type of input subject.

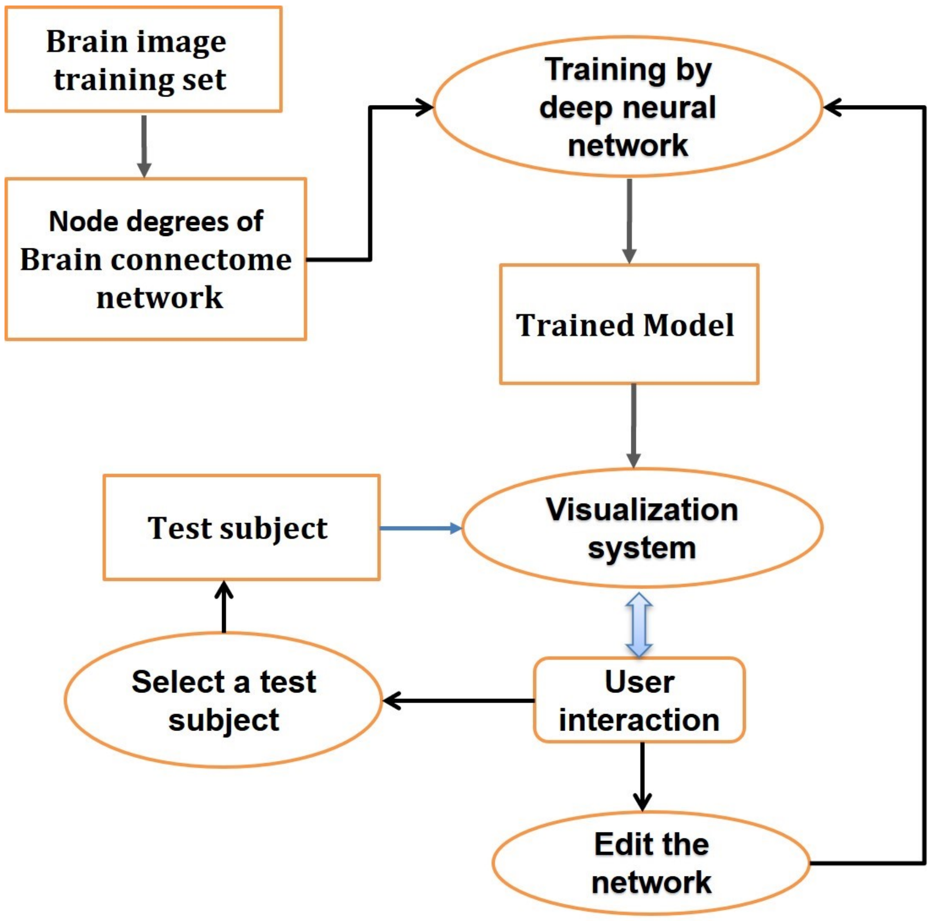


Fig. 4.27. Overall system flowchart.

Within our visualization system, the user can pick any node in the intermediate layers to show what the node represents in terms of weights of different brain regions. This provides the required contextual information to interpret the intermediate hidden layers. For example, when a brain region is consistently weighted high by some nodes in every layer, this could be an indication that this region is an important image phenotype for the Alzheimer's Disease classification problem. On the other hand, when a certain brain region is known to be associated with Alzheimer's Disease (e.g. the Hippocampus), nodes that carry high weights on this region may have more significance in the hidden layers, and could be a candidate to manipulate (e.g.

increasing associated edge weights) when there is a need to re-train the mode to improve performance. To compute the weights of brain regions for some node S (we will call it source node) in a hidden layer for a given input subject, we use a recursive method to backpropagate the importance factors from any node in a hidden layer back to the nodes in the input layer, which represents the network degrees of the different regions of interest (ROIs). Let N_i^k denotes the is node in the ks layer, with layer number 0 being the input layer. Let W_{ij}^k be the edge weight between node N_i^k and node N_j^{k-1}, and A_i^k be the activation value of node N_i^k (for the given input subject). We define the importance factor of node N_j^{k-1} (relative to the source nde) as:

$$r_j^{k-1} = \sum_i R_{ij}^{k-1}$$

where R_{ij}^{k-1} is the importance factor along edge N_j^{k-1} N_i^k, and is calculated by:

$$R_{ij}^{k-1} = \frac{|W_{ij}^k| * A_j^{k-1}}{\sum_l (|W_{il}^k| * A_{jl}^{k-1})} \times r_i^k$$

For example, the importance factor of node I1 along the edge J1I1 in Figure 4.28 is calculated as:

$$\frac{|w_1| * a_1}{|w_1| * a_1 + |w_2| * a_2 + |w_3| * a_2} \times r_{J1}$$

Intuitively, the importance factor of a node represents how much this node contributes to the activation of the source node S. Starting from a source node S, the algorithm recursively computes an importance factor for every node in all possible paths from S back to the input layer. When a node can be reached by multiple paths, multiple importance factors will be computed, and their sum will be the final importance factor for this node. This process is recursively carried out backwards until reaching the input layer. At that point, we will have all r_i^0, which are the weights of all input brain regions for the given source node S. It is easy to prove that the sum of all final importance factors of all nodes in any given layer is 1.

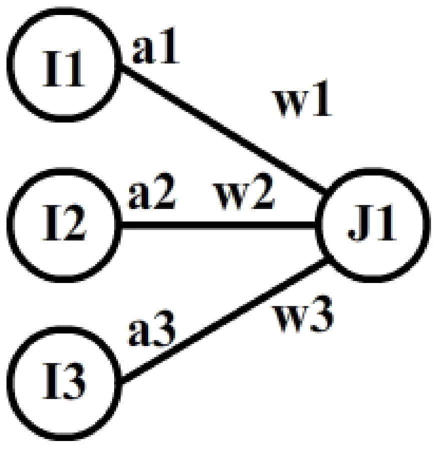


Fig. 4.28. An example of the importance backpropagation.

4.3.3 Interactions

The interactive component of our visualization framework consists of two types of interactions: (1) brain structure exploration; and (2) editing of the deep neural network structure.

In brain structure exploration, the user will pick a node in a hidden layer. The brain rendering window will show the color-coded 3D rendering of the brain as well as the 2D color-coded brain map. Rotation and zooming can be used for the user to examine and understand which brain regions are activated by this node. By examining multiple nodes in different layers, the user will gain understanding about how different brain regions are affected by the evolution of the neural networks to draw a conclusion for different labelled test subjects (e.g. an AD patient vs a control). It may also provide clues about why certain test subjects are classified incorrectly. This could potentially lead to changes to the network structure for re-training.

The edit functions are designed to allow the user to make changes to nodes and edges based on user observations and domain knowledge. After changes are made, the deep neural network model will need to be re-trained using the revised network structure and weights as the new initial condition. Three types of edit operators are defined in our system:

- Deleting a node
- Adding a new node
- Changing the weight of an edge

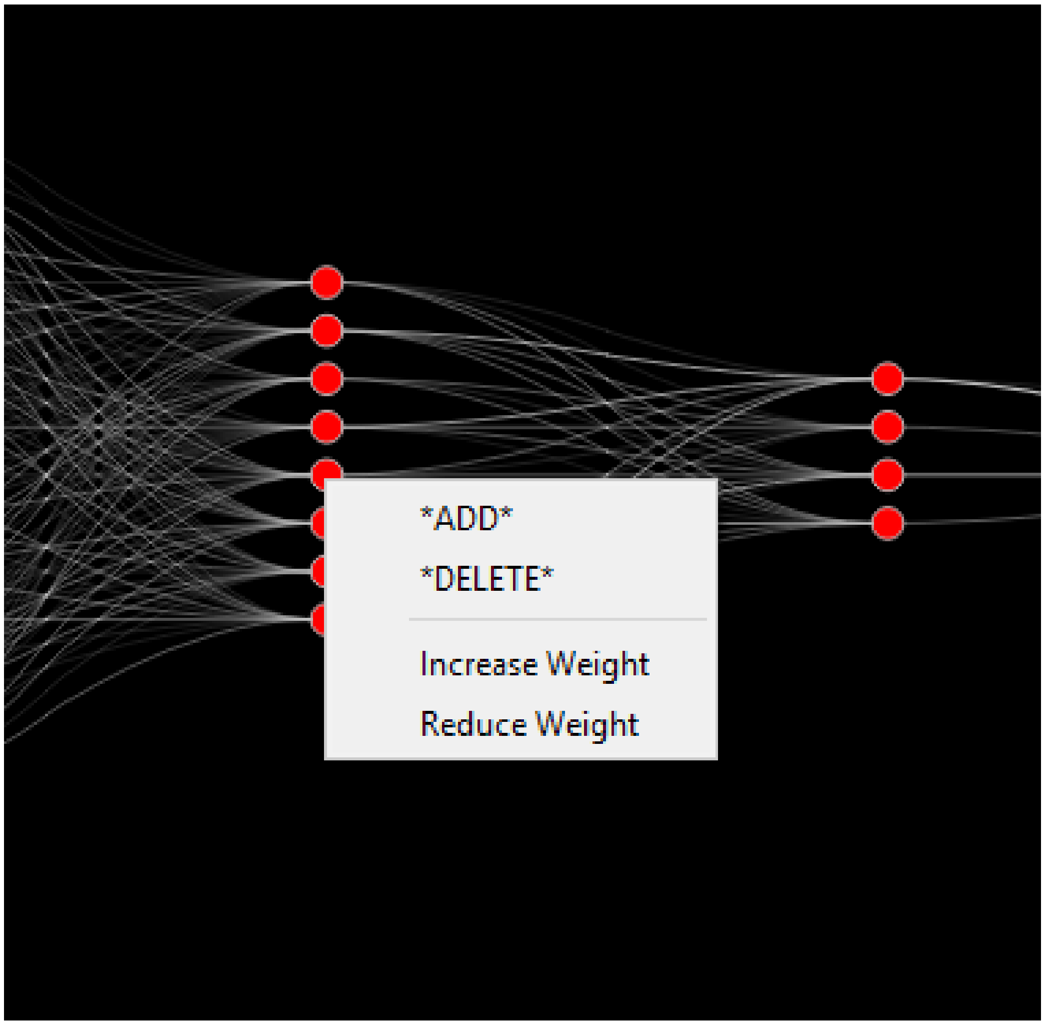
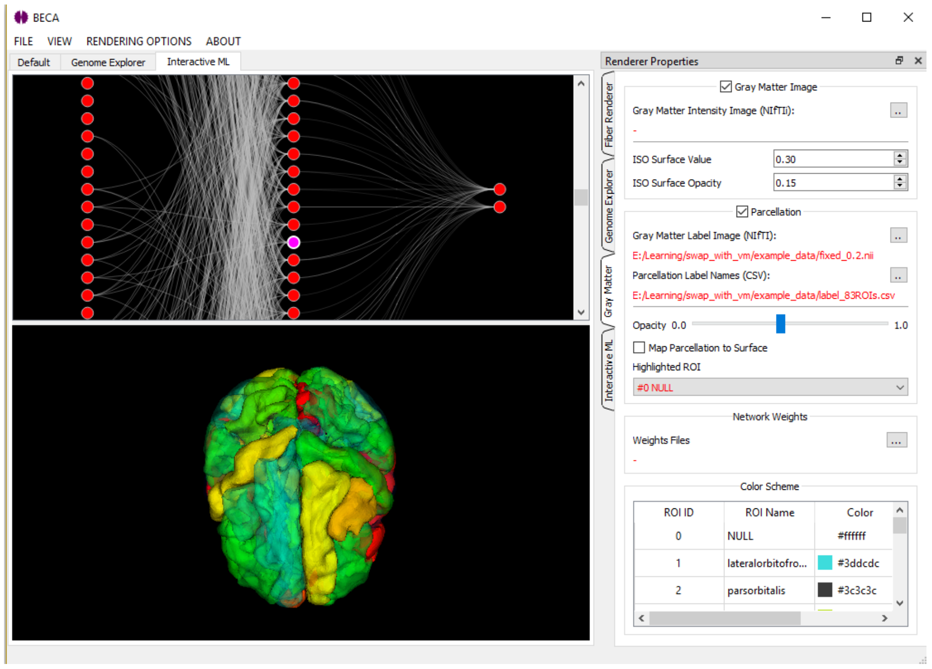


Fig. 4.29. User interface for editing a node.



Renderer Properties

[✓] Gray Matter Image

Gray Matter Intensity Image (NIfTIi):
-

ISO Surface Value 0.30

ISO Surface Opacity 0.15

[✓] Parcellation

Gray Matter Label Image (NIfTI):
E:/Learning/swap_with_vm/example_data/fixed_0.2.nii

Parcellation Label Names (CSV):
E:/Learning/swap_with_vm/example_data/label_83ROIs.csv

Opacity 0.0 1.0

[ ] Map Parcellation to Surface

Highlighted ROI
#0 NULL

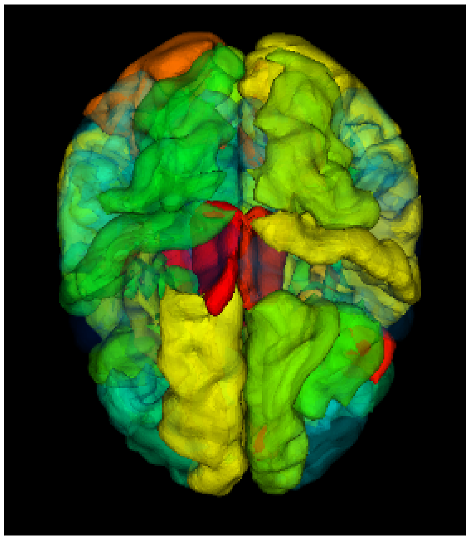
Network Weights

Weights Files
-

Color Scheme

ROI ID	ROI Name	Color
0	NULL	#ffffff
1	lateralorbitofro...	🟦 #3ddcdc
2	parsorbitalis	⬛ #3c3c3c

(a)



(b)

Fig. 4.30. (a) the interactive visualization system. (b) the discriminative ROIs for class AD

These operations can be carried out interactively through combinations of user mouse clicking and control panel editing. As an example, when a particular node has strong weight on a brain region that is known to be associated with Alzheimer's Disease, the user may decide to increase the weights of the edges that lead this node to the next layer. Another scenario is when the model does not perform well with internal test set, we may increase the number of nodes or even layers to generate more feature combinations.

From an optimization point of view, these changes can be considered an educated perturbation of the initial conditions of learning process. When a modified neural network structure is used as the initial condition, it can potentially move the system out of a local minimum.

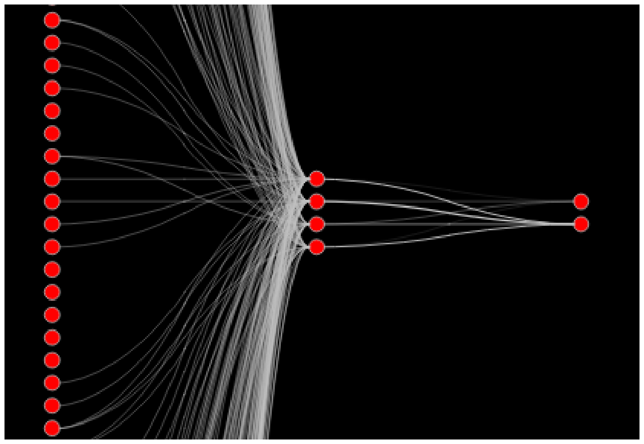
4.3.4 Experimental Results

As a pilot study, we trained a three-layer fully connected neural network as a classifier to distinguish healthy controls from patients diagnosed with Alzheimer's Disease (AD). These fibre density values are then used as the edge weights of the connectome network for each subject. The connectome network is now modelled as an undirected graph. We calculate the degree of each node (ROI) as the sum of weights of all connected edges to this node. This is our initial feature set for our deep learning analysis experiment.

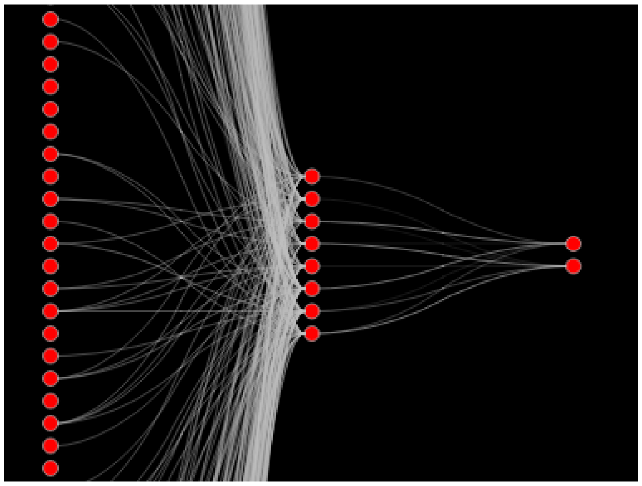
In our study, we focus on the analysis of the brain connectome data from 62 subjects in 2 categories: HC (Healthy Control, 43 subjects) and AD (Alzheimer's Disease, 19 subjects). The brain connectome network data from the 62 subjects are 62 graphs. Since all the subjects share the same parcellation, they have the same set of node labels (the brain ROIs) but different connectivity (edges) between the nodes as the fibre densities are different for different subjects between the same pairs of ROI labels

The inputs consist of 62 subjects. Each subject is a vector of 83 elements representing the fiber densities of 83 regions of interest (ROIs) in the brain. We built an interactive exploration system that allows user to click any of the nodes in the neural network to show the corresponding features with respect to the input layer calculated by our proposed importance backpropagation algorithm. We also mapped the importance of each input ROI on a 3D brain, to help neuroscientist to have an intuitive understanding of which ROIs are correlative with each class. The discriminative ROIs for class AD are shown in Figure 4.30(b). By visualizing the calculated importance of each ROI with a heatmap, we can see there are a small subset of the ROIs that is correlative with AD class, which are highlighted in red.

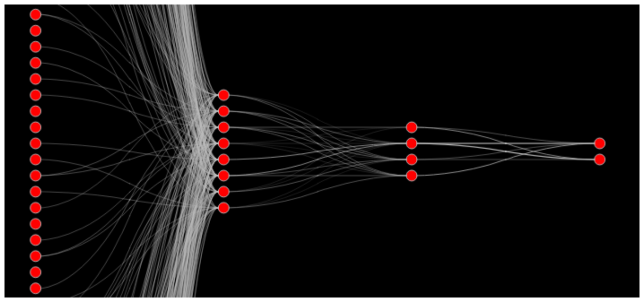
We also demonstrate the ability to change the neural network structure by interactively adding or deleting nodes with the system. As shown in Figure 4.31(a), we start with a three-layer fully connected neural network, in which the 2nd layer consist of 4 nodes. Then we add 4 extra nodes for the 2nd layer by right clicking at any node in the that layer, as shown in Figure 4.31(b). After re-training, the classification accuracy raises from 62.2% to 64.7%. We then add an extra layer of 4 nodes right before the output, which further boosts the classification results to 67.8%, which is shown in Figure 4.31(c).



(a) Network structure before adding nodes



(b) Network structure after adding 4 nodes in the second layer



(c) Network structure after adding extra layer with 4 nodes.

Fig. 4.31. Interactively adding nodes and layers. Classification accuracies are (a) 62.2%, (b) 64.7% and (c) 67.8%.

5. CONCLUSIONS

In this study, firstly, we have presented an integrated visualization solution for human brain data. Multiple modalities of images are involved including structural-MRI, funtinoal-MRI and DTI. Our focus is on the integration of analysis properties of the connectome networks into the anatomical brain structures. We apply a surface texture based approach to encode network properties and attributes onto the surfaces of the brain structures to establish visual connections and context. Surface texture is an effective approach to integrate information visualization and scientific visualization since scientific data typically have spatial structures containing surface areas, which can be taken advantage of for visual encoding. In the future, we would like to continue developing the integrated visualization tool for public domain distribution. Currently, a prototype BECA software tool is available at `http://www.iu.edu/~beca/`, and we will continue improving it. We would also like to study interesting visual analytic topics to compare multiple networks from different network construction procedures, in particular, between structural networks and functional networks.

Secondly, We have presented a general framework for a visualization supported interactive machine learning approach, and have tested the framework on two different types of application problems using real world datasets: handwriting recognition and human cognitive score prediction. The experiments show that interactively selected training samples can reach high performance quicker than randomly selected samples. This approach provides a new way to train a machine learning model using a small set of training samples. Since human knowledge and perceptual instincts are used in the selection of the training samples, this approach is potentially smarter and more efficient than traditional big data solutions. It is particularly useful for applications where high quality big data is not readily available or if the collection and labeling of the data is too expensive (e.g. in some biomedical data analysis applications). On the other hand, since this approach requires human in the learning loop, it may not be suitable for applications that require total automation (e.g. in real time robot vision). Although this paper focuses on the small data solution, the visual analytics framework proposed here can be applied to other types of interactive machine learning problems such as human knowledge integration and model optimization. In the future, we would like to explore new solutions to interactive machine learning with knowledge (e.g. rules and constraints) input, parameter settings and other model optimization functions. We would also like to investigate ways to automatically evaluate the models so that the additional samples can be added automatically. The proposed interactive visualization framework for a deep neural network system with brain image data. We have tested the system on a brain classification problem for the diagnosis of Alzheimer's Disease using a real world ADNI dataset. While the concept of visualizing deep neural networks to help user understand the learning model has been discussed before, we believe the key is to provide contextual information specific for the underlying applications. Our work focuses on building visual associations between deep neural network nodes and the complex brain structures. The backpropagation algorithm for computing the importance factors is efficient and easy to implement. Our system provides the needed biological and anatomical interpretation of the evolution of the deep neural network. Furthermore, it allows the users to make changes to the hidden layer nodes and edges based on their visual observations and domain knowledge to potentially improve the systems performance. This can be particularly useful for applications that require highly specialized knowledge and domain expertise (e.g. biomedical applications). We also have presented an interactive visualization framework for a deep neural network system with brain image data. We have tested the system on a brain classification problem for the diagnosis of Alzheimer's Disease using a real world ADNI dataset. While the concept of visualizing deep neural networks to help user understand the learning model has been discussed before, we believe the key is to provide contextual information specific for the underlying applications. Our work focuses on building visual associations between deep neural network nodes and the complex brain structures. The backpropagation algorithm for computing the importance factors is efficient and easy to implement. Our system provides the needed biological and anatomical interpretation of the evolution of the deep neural network. Furthermore, it allows the users to make changes to the hidden layer nodes and edges based on their visual observations and domain knowledge to potentially improve the systems performance. This can be particularly useful for applications that require highly specialized knowledge and domain expertise (e.g. biomedical applications). User interference and re-direction can also be helpful when automatic deep learning performs poorly. Although this paper focuses on the brain data classification problem, the visual analytics framework proposed here can be applied to a wide range of potential applications using deep learning techniques. In the future, we would like to develop more sophisticated interaction tools to facilitate better knowledge input and user feedback. We would also like to work with domain experts to collect more user experience to better evaluate the benefits of such a system.

REFERENCES

[1] M. H. Lee, C. D. Smyser, and J. S. Shimony, "Resting-state fmri: a review of methods and clinical applications," *American Journal of Neuroradiology*, vol. 34, no. 10, pp. 1866–1872, 2013.

[2] V. Petrovic, J. Fallon, and F. Kuester, "Visualizing whole-brain dti tractography with gpu-based tuboids and lod management," *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1488–1495, 2007.

[3] C. Stoll, S. Gumhold, and H.-P. Seidel, "Visualization with stylized line primitives," in *VIS 05. IEEE Visualization, 2005.* IEEE, 2005, pp. 695–702.

[4] D. Merhof, M. Sonntag, F. Enders, C. Nimsky, P. Hastreiter, and G. Greiner, "Hybrid visualization for white matter tracts using triangle strips and point sprites," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1181–1188, 2006.

[5] T. Peeters, A. Vilanova, and R. ter Haar Romeny, "Visualization of dti fibers using hair-rendering techniques," in *Proc ASCI*, 2006, pp. 66–73.

[6] G. J. Parker, H. A. Haroon, and C. A. Wheeler-Kingshott, "A framework for a streamline-based probabilistic index of connectivity (pico) using a structural interpretation of mri diffusion measurements," *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 18, no. 2, pp. 242–254, 2003.

[7] A. von Kapri, T. Rick, S. Caspers, S. B. Eickhoff, K. Zilles, and T. Kuhlen, "Evaluating a visualization of uncertainty in probabilistic tractography," in *Medical Imaging 2010: Visualization, Image-Guided Procedures, and Modeling*, vol. 7625. International Society for Optics and Photonics, 2010, p. 762534.

[8] S. Achard, R. Salvador, B. Whitcher, J. Suckling, and E. Bullmore, "A resilient, low-frequency, small-world human brain functional network with highly connected association cortical hubs," *Journal of Neuroscience*, vol. 26, no. 1, pp. 63–72, 2006.

[9] R. Salvador, J. Suckling, C. Schwarzbauer, and E. Bullmore, "Undirected graphs of frequency-dependent functional connectivity in whole brain networks," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, pp. 937–946, 2005.

[10] A. J. Schwarz and J. McGonigle, "Negative edges and soft thresholding in complex network analysis of resting state functional connectivity data," *Neuroimage*, vol. 55, no. 3, pp. 1132–1146, 2011.

[11] J. D. Van Horn, A. Irimia, C. M. Torgerson, M. C. Chambers, R. Kikinis, and A. W. Toga, "Mapping connectivity damage in the case of phineas gage," *PloS one*, vol. 7, no. 5, p. e37454, 2012.

[12] R. Schurade, M. Hlawitschka, B. Hamann, G. Scheuermann, T. R. Knösche, and A. Anwander, "Visualizing white matter fiber tracts with optimally fitted curved dissection surfaces," in *Proceedings of the 2nd Eurographics conference on Visual Computing for Biology and Medicine.* Eurographics Association, 2010, pp. 41–48.

[13] S. Eichelbaum, M. Hlawitschka, and G. Scheuermann, "Lineaoimproved three-dimensional line rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 433–445, 2012.

[14] P. Svetachov, M. H. Everts, and T. Isenberg, "Dti in context: Illustrating brain fiber tracts in situ," in *Computer Graphics Forum*, vol. 29, no. 3. Wiley Online Library, 2010, pp. 1023–1032.

[15] K. A. Cook and J. J. Thomas, "Illuminating the path: The research and development agenda for visual analytics," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2005.

[16] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[17] J. Foucher, P. Vidailhet, S. Chanraud, D. Gounot, D. Grucker, D. Pins, C. Damsa, and J.-M. Danion, "Functional integration in schizophrenia: too little or too much? preliminary results on fmri data," *Neuroimage*, vol. 26, no. 2, pp. 374–388, 2005.

[18] K. J. Worsley, J.-I. Chen, J. Lerch, and A. C. Evans, "Comparing functional connectivity via thresholding correlations and singular value decomposition," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1457, pp. 913–920, 2005.

[19] X.-N. Zuo, R. Ehmke, M. Mennes, D. Imperati, F. X. Castellanos, O. Sporns, and M. P. Milham, "Network centrality in the human functional connectome," *Cerebral cortex*, vol. 22, no. 8, pp. 1862–1875, 2011.

[20] E. R. Tufte, "The visual display of quantitative information graphics press," *Cheshire, Connecticut*, 1983.

[21] M. Weber, M. Alexa, and W. Müller, "Visualizing time-series on spirals." in *Infovis*, vol. 1, 2001, pp. 7–14.

[22] S. Havre, B. Hetzler, and L. Nowell, "Themeriver: Visualizing theme changes over time," in *IEEE Symposium on Information Visualization 2000. INFOVIS 2000. Proceedings.* IEEE, 2000, pp. 115–123.

[23] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong, "Textflow: Towards better understanding of evolving topics in text," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2412–2421, 2011.

[24] M. D. Fox, A. Z. Snyder, J. L. Vincent, M. Corbetta, D. C. Van Essen, and M. E. Raichle, "The human brain is intrinsically organized into dynamic, anticorrelated functional networks," *Proceedings of the National Academy of Sciences*, vol. 102, no. 27, pp. 9673–9678, 2005.

[25] C. Chang and G. H. Glover, "Time–frequency dynamics of resting-state brain connectivity measured with fmri," *Neuroimage*, vol. 50, no. 1, pp. 81–98, 2010.

[26] D. A. Handwerker, V. Roopchansingh, J. Gonzalez-Castillo, and P. A. Bandettini, "Periodic changes in fmri connectivity," *Neuroimage*, vol. 63, no. 3, pp. 1712–1719, 2012.

[27] E. A. Allen, E. Damaraju, S. M. Plis, E. B. Erhardt, T. Eichele, and V. D. Calhoun, "Tracking whole-brain connectivity dynamics in the resting state," *Cerebral cortex*, vol. 24, no. 3, pp. 663–676, 2014.

[28] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," Lawrence Livermore National Lab., CA (United States), Tech. Rep., 1993.

[29] D. Stalling and H.-C. Hege, "Fast and resolution independent line integral convolution," 1995.

[30] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf, "The state of the art in flow visualization: Dense and texture-based techniques," in *Computer Graphics Forum*, vol. 23, no. 2. Wiley Online Library, 2004, pp. 203–221.

[31] T. Mcgraw and M. Nadar, "Fast texture-based tensor field visualization for dt-mri," in *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, 2007, pp. 760–763.

[32] C. Auer, C. Stripf, A. Kratz, and I. Hotz, "Glyph-and texture-based visualization of segmented tensor fields." in *GRAPP/IVAPP*, 2012, pp. 670–677.

[33] M. Levoy, "Display of surfaces from volume data: Ieee comput. graph. & appl. vol 8 no 3 (1988) pp 29–37," 1988.

[34] K. Kreeger, I. Bitter, F. Dachille, B. Chen, and A. Kaufman, "Adaptive perspective ray casting," in *IEEE Symposium on Volume Visualization (Cat. No. 989EX300)*. IEEE, 1998, pp. 55–62.

[35] A. Knoll, Y. Hijazi, R. Westerteiger, M. Schott, C. Hansen, and H. Hagen, "Volume ray casting with peak finding and differential sampling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1571–1578, 2009.

[36] M. Levoy, "Efficient ray tracing of volume data," *ACM Transactions on Graphics (TOG)*, vol. 9, no. 3, pp. 245–261, 1990.

[37] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler, "The volumepro real-time ray-casting system," 1999.

[38] K. Mueller and R. Yagel, "Fast perspective volume rendering with splatting by utilizing a ray-driven approach," in *Proceedings of Seventh Annual IEEE Visualization'96*. IEEE, 1996, pp. 65–72.

[39] L. Westover, “Footprint evaluation for volume rendering,” *ACM Siggraph Computer Graphics*, vol. 24, no. 4, pp. 367–376, 1990.

[40] ——, “Interactive volume rendering,” in *Proceedings of the 1989 Chapel Hill workshop on Volume visualization.* ACM, 1989, pp. 9–16.

[41] L. A. Westover, “Splatting: a parallel, feed-forward volume rendering algorithm,” Ph.D. dissertation, University of North Carolina at Chapel Hill Chapel Hill, NC, 1991.

[42] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, “Ewa volume splatting,” in *Proceedings Visualization, 2001. VIS’01.* IEEE, 2001, pp. 29–538.

[43] P. Lacroute and M. Levoy, “Fast volume rendering using a shear-warp factorization of the viewing transformation,” in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques.* ACM, 1994, pp. 451–458.

[44] J. Kruger and R. Westermann, “Acceleration techniques for gpu-based volume rendering,” in *Proceedings of the 14th IEEE Visualization 2003 (VIS’03).* IEEE Computer Society, 2003, p. 38.

[45] S. Stegmaier, M. Strengert, T. Klein, and T. Ertl, “A simple and flexible volume rendering framework for graphics-hardware-based raycasting,” in *Fourth International Workshop on Volume Graphics, 2005.* IEEE, 2005, pp. 187–241.

[46] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *ACM siggraph computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.

[47] M. Hadwiger, C. Sigg, H. Scharsach, K. Bühler, and M. Gross, “Real-time ray-casting and advanced shading of discrete isosurfaces,” in *Computer graphics forum*, vol. 24, no. 3. Wiley Online Library, 2005, pp. 303–312.

[48] M. Šrámek, “Fast surface rendering from raster data by voxel traversal using chessboard distance,” in *Proceedings of the conference on Visualization’94.* IEEE Computer Society Press, 1994, pp. 188–195.

[49] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten, “Interactive machine learning: letting users build classifiers,” *International Journal of Human-Computer Studies*, vol. 55, no. 3, pp. 281–292, 2001.

[50] J. G. Paiva, L. Florian, H. Pedrini, G. Telles, and R. Minghim, “Improved similarity trees and their application to visual data classification,” *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2459–2468, 2011.

[51] J. Xia, W. Chen, Y. Hou, W. Hu, X. Huang, and D. S. Ebertk, “Dimscanner: A relation-based visual exploration approach towards data dimension inspection,” in *2016 IEEE Conference on Visual Analytics Science and Technology (VAST).* IEEE, 2016, pp. 81–90.

[52] S. Liu, X. Wang, M. Liu, and J. Zhu, “Towards better analysis of machine learning models: A visual analytics perspective,” *Visual Informatics*, vol. 1, no. 1, pp. 48–56, 2017.

[53] T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the black box: Understanding dqns," in *International Conference on Machine Learning*, 2016, pp. 1899–1908.

[54] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea, "Visualizing the hidden activity of artificial neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 101–110, 2016.

[55] F.-Y. Tzeng and K.-L. Ma, "Opening the black box-data driven visualization of neural networks," in *VIS 05. IEEE Visualization, 2005.* IEEE, 2005, pp. 383–390.

[56] A. W. Harley, "An interactive node-link visualization of convolutional neural networks," in *International Symposium on Visual Computing.* Springer, 2015, pp. 867–877.

[57] M. J. Streeter, M. O. Ward, and S. A. Alvarez, "Nvis: An interactive visualization tool for neural networks," in *Visual Data Exploration and Analysis VIII*, vol. 4302. International Society for Optics and Photonics, 2001, pp. 234–242.

[58] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 91–100, 2016.

[59] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[60] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *arXiv preprint arXiv:1702.04595*, 2017.

[61] S. Lim, "A light-weight visualization tool for support vector machines," in *2014 25th International Workshop on Database and Expert Systems Applications.* IEEE, 2014, pp. 94–98.

[62] L. Hamel, "Visualization of support vector machines with unsupervised learning," in *2006 IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology.* IEEE, 2006, pp. 1–8.

[63] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting interactive performance analysis for multiclass classifiers," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 61–70, 2016.

[64] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber, "Visual methods for analyzing probabilistic classification data," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1703–1712, 2014.

[65] J. Chuang, S. Gupta, C. Manning, and J. Heer, "Topic model diagnostics: Assessing domain relevance via topical alignment," in *International Conference on Machine Learning*, 2013, pp. 612–620.

[66] J. Krause, A. Perer, and E. Bertini, "Using visual analytics to interpret predictive machine learning models," *arXiv preprint arXiv:1606.05685*, 2016.

[67] J. Krause, A. Perer, and K. Ng, "Interacting with predictions: Visual inspection of black-box machine learning models," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 5686–5697.

[68] J. Wang, "Multigraph visualization for feature classification of brain network data," Ph.D. dissertation, 2016.

[69] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim, "An approach to supporting incremental visual data classification," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 1, pp. 4–17, 2014.

[70] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan, "An uncertainty-aware approach for exploratory microblog retrieval," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 250–259, 2015.

[71] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 1992–2001, 2013.

[72] X. Wang, S. Liu, J. Liu, J. Chen, J. Zhu, and B. Guo, "Topicpanorama: A full picture of relevant topics," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2508–2521, 2016.

[73] T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the black box: Understanding dqns," in *International Conference on Machine Learning*, 2016, pp. 1899–1908.

[74] P. E. Rauber, S. G. Fadel, A. X. Falcao, and A. C. Telea, "Visualizing the hidden activity of artificial neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 101–110, 2016.

[75] F.-Y. Tzeng and K.-L. Ma, "Opening the black box-data driven visualization of neural networks," in *VIS 05. IEEE Visualization, 2005*. IEEE, 2005, pp. 383–390.

[76] A. W. Harley, "An interactive node-link visualization of convolutional neural networks," in *International Symposium on Visual Computing*. Springer, 2015, pp. 867–877.

[77] M. J. Streeter, M. O. Ward, and S. A. Alvarez, "Nvis: An interactive visualization tool for neural networks," in *Visual Data Exploration and Analysis VIII*, vol. 4302. International Society for Optics and Photonics, 2001, pp. 234–241.

[78] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 91–100, 2016.

[79] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[80] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," *arXiv preprint arXiv:1702.04595*, 2017.

[81] J. Wang, L. Gou, H.-W. Shen, and H. Yang, "Dqnviz: A visual analytics approach to understand deep q-networks," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 288–298, 2018.

[82] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg, "Gan lab: Understanding complex deep generative models using interactive visual experimentation," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 1–11, 2018.

[83] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[84] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[85] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting interactive performance analysis for multiclass classifiers," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 61–70, 2016.

[86] B. Alsallakh, A. Hanbury, H. Hauser, S. Miksch, and A. Rauber, "Visual methods for analyzing probabilistic classification data," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1703–1712, 2014.

[87] J. Chuang, S. Gupta, C. Manning, and J. Heer, "Topic model diagnostics: Assessing domain relevance via topical alignment," in *International Conference on Machine Learning*, 2013, pp. 612–620.

[88] S. Liu, X. Wang, M. Liu, and J. Zhu, "Towards better analysis of machine learning models: A visual analytics perspective," *Visual Informatics*, vol. 1, no. 1, pp. 48–56, 2017.

[89] J. Krause, A. Perer, and E. Bertini, "Using visual analytics to interpret predictive machine learning models," *arXiv preprint arXiv:1606.05685*, 2016.

[90] J. Krause, A. Perer, and K. Ng, "Interacting with predictions: Visual inspection of black-box machine learning models," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 5686–5697.

[91] J. Wang, "Multigraph visualization for feature classification of brain network data," Ph.D. dissertation, 2016.

[92] J. G. S. Paiva, W. R. Schwartz, H. Pedrini, and R. Minghim, "An approach to supporting incremental visual data classification," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 1, pp. 4–17, 2014.

[93] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan, "An uncertainty-aware approach for exploratory microblog retrieval," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 250–259, 2015.

[94] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 12, pp. 1992–2001, 2013.

[95] S. Liu, X. Wang, J. Chen, J. Zhu, and B. Guo, "Topicpanorama: A full picture of relevant topics," in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2014, pp. 183–192.

[96] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, "Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 364–373, 2018.

[97] D. Sacha, M. Kraus, D. A. Keim, and M. Chen, "Vis4ml: An ontology for visual analytics assisted machine learning," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 385–395, 2018.

[98] H. Li, S. Fang, S. Mukhopadhyay, A. J. Saykin, and L. Shen, "Interactive machine learning by visualization: A small data solution," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 3513–3521.

[99] P. Hagmann, L. Cammoun, X. Gigandet, R. Meuli, C. J. Honey, V. J. Wedeen, and O. Sporns, "Mapping the structural core of human cerebral cortex," *PLoS biology*, vol. 6, no. 7, p. e159, 2008.

[100] P. Hagmann, M. Kurant, X. Gigandet, P. Thiran, V. J. Wedeen, R. Meuli, and J.-P. Thiran, "Mapping human whole-brain structural networks with diffusion mri," *PloS one*, vol. 2, no. 7, p. e597, 2007.

[101] J. D. Power, A. Mitra, T. O. Laumann, A. Z. Snyder, B. L. Schlaggar, and S. E. Petersen, "Methods to detect, characterize, and remove motion artifact in resting state fmri," *Neuroimage*, vol. 84, pp. 320–341, 2014.

[102] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

[103] B. B. Biswal, "Resting state fmri: a personal history," *Neuroimage*, vol. 62, no. 2, pp. 938–944, 2012.

[104] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern recognition letters*, vol. 1, no. 2, pp. 95–102, 1982.

[105] N. Gossett and B. Chen, "Paint inspired color mixing and compositing for visualization," in *IEEE symposium on information visualization*. IEEE, 2004, pp. 113–118.

[106] Y. Liang *et al.*, "Brain connectome visualization for feature classification," in *Proceedings of IEEE visualization*, 2014.

[107] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[108] J. P. Snyder, *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.

[109] Z. C. Lipton, "The mythos of model interpretability," *arXiv preprint arXiv:1606.03490*, 2016.

[110] K. A. Cook and J. J. Thomas, "Illuminating the path: The research and development agenda for visual analytics," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2005.

[111] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Magazine*, vol. 35, no. 4, pp. 105–120, 2014.