Summer 2016

Machine Learning Methods for Medical and Biological Image Computing

Rongjian Li
Old Dominion University

MACHINE LEARNING METHODS FOR MEDICAL AND BIOLOGICAL IMAGE COMPUTING

by

Rongjian Li
B.S. July, 2007, Xi'an Jiaotong University, China
M.S. July 2012, Xi'an Jiaotong University, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2016

Approved by:

Shuiwang Ji (Director)

Andrey Chernikov (Member)

Jing He (Member)

Jiang Li (Member)

ABSTRACT

MACHINE LEARNING METHODS FOR MEDICAL AND BIOLOGICAL IMAGE COMPUTING

Rongjian Li
Old Dominion University, 2016
Director: Dr. Shuiwang Ji

Medical and biological imaging technologies provide valuable visualization information of structure and function for an organ from the level of individual molecules to the whole object. Brain is the most complex organ in body, and it increasingly attracts intense research attentions with the rapid development of medical and biological imaging technologies. A massive amount of high-dimensional brain imaging data being generated makes the design of computational methods for efficient analysis on those images highly demanded. The current study of computational methods using hand-crafted features does not scale with the increasing number of brain images, hindering the pace of scientific discoveries in neuroscience. In this thesis, I propose computational methods using high-level features for automated analysis of brain images at different levels. At the brain function level, I develop a deep learning based framework for completing and integrating multi-modality neuroimaging data, which increases the diagnosis accuracy for Alzheimer's disease. At the cellular level, I propose to use three dimensional convolutional neural networks (CNNs) for segmenting the volumetric neuronal images, which improves the performance of digital reconstruction of neuron structures. I design a novel CNN architecture such that the model training and testing image prediction can be implemented in an end-to-end manner. At the molecular level, I build a voxel CNN classifier to capture discriminative features of the input along three spatial dimensions, which facilitate the identification of secondary structures of proteins from electron microscopy images. In order to classify genes specifically expressed in different brain cell-type, I propose to use invariant image feature descriptors to capture local gene expression information from cellular-resolution *in situ* hybridization images. I build image-level representations by applying regularized learning and vector quantization on generated image descriptors. The developed computational methods in this dissertation are evaluated using images from medical and biological experiments in comparison with baseline methods. Experimental results demonstrate that the developed representations, formulations, and algorithms are effective and efficient in learning from brain imaging data.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Prof. Shuiwang Ji, for his guidance, encouragement, and support during my research. He is an incredible advisor, an enthusiastic researcher, and an easygoing friend. The experience with him is my lifelong assets. I am deeply grateful for my committee members, Prof. Andrey Chernikov, Prof. Jing He and Prof. Jiang Li, for their great help throughout my Ph.D studies. I am very grateful for all your insights and advice during the past four years.

During my Ph.D study, I have been very fortunate to work with a group of incredible people to whom I deeply indebted. I would like to thank my collaborators, Wenlu Zhang, Tao Zeng, Ahmed Fakhry and Dong Si for their insightful discussions and valuable interactions.

Last but not least, I want to thank my family for their support and understanding.

My dissertation work is supported in part by the National Science Foundation grant DBI-1147134, DBI-1350258 and DBI-1356621.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

Images play an increasingly important role in many scientific fields. Biology is one of the best examples that depend heavily upon images for progress. With the recent development of fluorescent probes and new high-resolution microscopes in biological imaging, researchers now can visualize sub-cellular components and processes both structurally and functionally in two or three dimensions. In particular, the increasing availability in biological images has led to intense research interests to the study of brain. In addition, medical imaging has become a powerful tool to understand how the brain processes sensory information, and how the processing mechanism ultimately leads to certain behavior or function. However, the currently used data analysis techniques for medical and biological imaging data are still relatively crude. Hence, in this thesis, I mainly work on the design of efficient machine learning methods for analyzing medical and biological images, and finally promoting the scientific discoveries about brain.

Brain is an extremely important and complex organ in human body. It is composed of organized cells through effective and functional circuits to control and coordinate actions and reactions of the body. The study of brain is usually proceeded at multiple levels, ranging from the molecular level, cellular level to the system and cognitive level. With rapid development of computers and vast amount of data generated by brain imaging techniques, machine learning is becoming more and more important for extracting reliable and meaningful information and making accurate predictions on brain imaging data. In this thesis, I focus on the design of efficient machine learning methods for analyzing multiple kinds of brain images at different levels. The challenges and contributions of my research are provided in the following.

At the brain function level, neuroimaging is a very powerful tool for research and medical diagnosis of brain conditions since it enables noninvasive visualization of the structure and functionality of the brain. There has been a number of successful neuroimaging techniques, such as magnetic resonance images (MRI) and positron emission tomography (PET) images and so on, for facilitating medical diagnosis. Combining multi-modality brain imaging data for disease diagnosis commonly leads to better performance. A key challenge in fully exploiting multiple imaging modalities is that the data are commonly incomplete; namely, some modality might be missing for some subjects. Therefore, in this thesis, I propose to use deep learning methods to complete multi-modality medical imaging data and then improve diagnose performance.

At the cellular-level, the study of structures and functions of neurons and neuron networks is crucial for understanding the working mechanisms of the brain. A key component in this process is the extraction of neuronal morphology from microscopic imaging data. Digital reconstruction, or tracing, of 3D neuron structure from microscopy images is a critical step toward reversing engineering the wiring and anatomy of a brain. Despite a number of prior attempts, this task remains very challenging, especially when images are contaminated by noises or have discontinued segments of neurite patterns. An approach for addressing such problems is to identify the locations of neuronal voxels using image segmentation methods prior to applying tracing or reconstruction techniques. This preprocessing step is expected to remove noises in the data, thereby leading to improved reconstruction results. Therefore, in this thesis, I propose 3D convolutional neural networks (CNNs) for segmenting neuronal microscopy images to improve neuron morphology reconstructions.

At the molecular level, the study of nervous system concerns the molecular basis of biological activities between bio-molecules in neurons, including deoxyribonucleic acids (DNAs), ribonucleic acids (RNAs) and proteins, as well as their interactions. The fundamental research topics include development mechanisms of neurons and effects of genetic changes on nervous system. The biological discoveries in last century show that development of a multicellular organism is regulated by gene expression patterns that specify which part of DNA is used to synthesize proteins in each cell type. Differential gene expression patterns in cells of the mammalian brain result in the morphological, connectional, and functional diversity of cells. A wide variety of studies have shown that certain genes are expressed only in specific cell-types. Analysis of cell-type-specific gene expression patterns can provide insights into the relationship between genes, connectivity, brain regions, and cell-types. However, automated methods for identifying cell-type-specific genes are lacking to date. Therefore, in this thesis, I propose computational methods to identify cell-type-specific genes in the mouse brain by automated biological image computing of *in situ* hybridization (ISH) expression patterns.

Proteins are the major structural and functional components of cells. The study about folded structures of proteins is of high importance for understanding the working mechanisms of actions in large macromolecular complexes such as neurons. However, the detection of secondary structure of proteins using 3 dimensional (3D) electron microscopy (EM) images is still a challenging task, especially when the spatial resolution of the EM images is at medium level (5-10Å). Prior researches focused on the usage of local features with image processing techniques to detect positions and shapes of proteins in 3D space. These methods usually involve human interventions but with limited success since the features used are hand-crafted which did not capture the global information of image objects. Therefore, in this thesis, I propose to use deep learning methods for extracting high representative global features and then automatically detecting secondary structures of proteins.

1.1 CONTRIBUTIONS OF THIS DISSERTATION

1. I propose a deep learning based framework for completing multimodality imaging data. The proposed deep learning method takes the form of convolutional neural networks, where the input and output are two volumetric modalities. The network contains a large number of trainable parameters that capture the highly complicated relationship between the input and output modalities. When trained on subjects with both modalities, the network is able to predict the output modality given the input modality, thereby allowing me to complete multi-modality data.

2. I propose to use 3D Convolutional neural networks (CNNs) for segmenting the

neuronal microscopy images. I design a novel CNN architecture that takes volumetric images as the inputs and their voxel-wise segmentation maps as the outputs. The developed architecture allows me to train and predict using large microscopy images in an end-to-end manner. In addition, to reduce the computational complexity for predictions of large images, I employ a stack of deconvolution layers in the CNN architecture that can produce a dense pixel-wise prediction very efficiently. I show that the proposed method improves the tracing performance significantly when combined with different reconstruction algorithms.

3. I build a voxel CNN classifier that predicts the probability of every individual voxel in an EM image with respect to the secondary structure elements of proteins such as α-helix, β-sheet and background. To effectively incorporate the 3D spatial information in protein structures, I propose to perform 3D convolution in the convolutional layers of CNNs so that discriminative features along three spatial dimensions are all captured. I show that the proposed CNN classifier can outperform existing learning based methods on identifying the secondary structure elements of proteins.

4. I propose to use invariant image feature descriptors to capture local gene expression information from cellular-resolution ISH images. I then build image-level representations by applying vector quantization on the image descriptors. I employ regularized learning methods for classifying genes specifically expressed in different brain cell-types. These methods can also rank image features based on their discriminative power. I show that the proposed automated image computing methods could potentially be used to identify cell-type-specific genes in the mouse brain.

1.2 SUMMARY OF REMAINING CHAPTERS

Chapter 2: Deep Learning for medical image completion. In this chapter, I first discuss the Alzheimers Disease Neuroimaging Initiative database and the properties of images in the database are presented. I then introduce the traditional convolutional neural networks widely applied for two dimensional natural images. Next, I describe the developed 3D convolutional neural networks for the imaging data completion. I also present the detailed architectures in which the organization of different types of layers for training is discussed. Finally, I report experimental results and discuss the broad impacts of the developed model for disease diagnosis on medical images.

Chapter 3: Deep learning for 3d neuron reconstruction. This chapter presents deep learning methods for neuron tracing problems. I first discuss the convolutional neural network framework, in which the challenges of applying CNNs over large volumetric data are introduced. I then present the proposed scheme for fast prediction of testing images of large size. Furthermore, I present the proposed CNN architectures in which novel designs are discussed. Finally, I report experimental results and discuss the broad impacts of improving the 3D neuron reconstruction.

Chapter 4: Deep learning for protein structure detection. This chapter presents the deep learning method for detecting the secondary structure elements of proteins on EM images. I first discuss the deep learning method briefly, in which the inception learning, residual learning and also dilated convolutions are introduced. I then present the proposed deep learning model for classifying the EM image voxels.

I report experimental results of my method on simulated EM images with medium resolution in terms of Cα atoms identification accuracy and secondary structure detection accuracy.

Chapter 5: Identification of cell-type-specific genes. In this chapter, I first discuss the *in situ* hybridization (ISH) image feature extraction scheme, in which the properties of scale-invariant feature transform are introduced. I then describe the developed computational models for cell-type-specific gene classification. Next, I present the feature selection formalism using stability selection scheme. Finally, I evaluate the experimental performance of my method on Allen Mouse Brain Atlas image data in terms of classification accuracy.

Chapter 6: Conclusion and outlook. In this chapter, I provide a summary of my contributions and discuss future research directions.

CHAPTER 2

DEEP LEARNING FOR MEDICAL IMAGE COMPLETION

2.1 INTRODUCTION AND RELATED WORK

Alzheimer's disease (AD) is a common neuro-degenerative disease. The diagnose of Alzheimer's disease is of great importance for early preventions and on-time treatments for patients. During the whole progression procedure of Alzheimer's disease, there are two important stages which are widely used to monitor the speed of disease evolution over time. These two stages are called stable mild cognitive impairment (sMCI) and progressive mild cognitive impairment (pMCI). The discrimination of these two stages is a challenge task since the characteristics of pMCI and sMCI are quite similar. The detection of pMCI and sMCI was initially implemented by experienced experts through observations. Recently, advanced computational methods coupled with multi-modal neuroimaging techniques have been developed for early diagnosis of AD or MCI [1].

A key challenge in employing computational methods for Alzheimer's disease diagnosis is that the neuroimaging data is usually incomplete in the sense that not all subjects have all data modalities. For example, the individual positron emission tomography (PET) scans are more expensive than conventional magnetic resonance imaging (MRI), thus some subjects might only have done the MRI scans because of the high cost. However, it would be beneficial if the data could be completed as different modalities contain complementary information. Nevertheless, the relationship between different data modalities is complicated and nonlinear. Thus, a highly sophisticated model is required for the collaborative completion of neuroimaging data.

In this dissertation, I propose to use deep CNNs for completing and integrating multi-modality neuroimaging data. Specifically, I designed a 3-dimensional (3D) CNN architecture that takes one volumetric data modality as input and another volumetric data modality as its output. When trained end-to-end on subjects with both data modalities, the network captures the nonlinear relationship between two data modalities and thus is able to predict the output data modality given the input modality. This allows me to complete and integrate incomplete data. To the best of my knowledge, this work represents the first attempt in using deep networks for data completion.

In this dissertation, I apply my 3-D CNN model to predict the missing PET patterns from the MRI data. I train my model on subjects with both PET and MRI data, where the MRI data were used as input and the PET data were used as output. The trained network contains a large number of parameters and thus is able to encode the nonlinear relationship between brain structure and function. I then use the trained network to predict the PET patterns for subjects with only MRI data. Results show that the predicted PET data achieved similar classification performance as the true PET images. Additionally, results show that my data completion method significantly outperforms prior methods in all experiments on disease diagnosis.

2.2 OVERVIEW OF CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNNs) are a class of deep learning models that attempt to compute high-level representations of data using multiple layers of nonlinear transformations. Prior studies have shown that CNNs achieved superior performance on object recognition [2] and classification tasks in natural images [3]. In addition, CNNs have been used in biological applications such as restoring and segmenting volumetric electron microscopy images [4–6].

CNNs mimic the visual information processing in the vision system of the brain by applying local filters to the input [7]. Such filters can be trained to extract various local features. To generate specific features of the entire visual field, a sliding filter is applied across the entire visual field. Such feature extraction method is also known as parameter sharing and leads to dramatic reduction in the number of trainable parameters. CNN models usually consist of alternating combination of convolutional layers and local neighborhood pooling layers, resulting in complex hierarchical representations of the inputs. These properties make CNNs a powerful tool in image-related applications.

A typical CNN is constructed by stacking various kinds of layers including convolutional layers, pooling layers, soft-max layers and so on. A convolutional layer uses a set of learnable filters to produce output. Each filter have a small receptive field, and is applied in a sliding way on input to generate an activation map called feature map. Feature maps give the responses of those filters at every spatial position. To be specific, if the k-th feature map at a given layer is h^k and the corresponding filter contains the weight W^k and the bias term b_k, then the output feature map h^k is obtained as follows:

$$h_{ij}^k = f((W^k * x)_{ij} + b_k).$$

Here, x is input vector from the previous layer, $*$ is the inner product of two vectors and f is a non-linear activation function. The commonly used form of f is the sigmoid function. Pooling layers are another types of layers used in CNNs. They generate output feature maps by applying filters without trainable parameters. The function of a pooling layer is to reduce the spatial size of output feature map, thus decrease the amount of parameters and computation in the network. A pooling layer also helps achieve invariance to visual distortions by ignoring positional information, which provides additional robustness for networks. Besides convolutional layers and pooling layers, a soft-max layer is also widely implemented as the final layer of a network used for many image related problems. Its function is to transform outputs of the previous layer to be in range $[0, 1]$ such that they have a probabilistic meaning. For example, for a K-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_K]$, the formula of soft-max function is given by

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}.$$

Besides the above mentioned layers, there are many other types of layers that have been proved very useful for obtaining competitive performance using CNNs. The detailed introductions about those layers are omitted in this dissertation, and can be found in the reference [8].

The training procedure for CNNs generally consists of two steps which are feed-forward computation and backpropagation. In feed-forward step, information moves in forward direction which is from input nodes, through hidden nodes, to output nodes. In backpropagation step, a loss between outputs of CNNs and ground truth is firstly computed. The gradient of loss function with respect to all weights and biases in network are then calculated. Weights and biases are updated through obtained gradient to minimize loss function. The widely used loss function is cross-entropy loss with following formula,

$$H(\mathbf{p}, \mathbf{q}) = -\sum_{i=1}^{K} p_i log(q_i)$$

where $\mathbf{p} = [p_1, p_2, \ldots, p_K]$ stands for ground truth vector and $\mathbf{q} = [q_1, q_2, \ldots, q_K]$ is for output vector of a CNN through feed forward computation. The training of CNNs is realized by alternatively implementing these two steps until stopping criteria is satisfied.

Note that there are two ways in backpropagation step for updating weights of CNNs. The first way is called batch learning in which updating is applied only after all training samples are visited and their gradient contributions are accumulated. The major advantage of batch learning is training could be converged in limited iterations and the global minimum of corresponding optimization problem could be reached. However, the computation load of batch learning will be high when the number of samples is very large. Therefore, another way of updating weights of CNNs, called online learning, is mainly used for real world applications. In online learning, gradients are calculated after using a subset of training samples to update weights for each iteration. The convergence of online learning is slower but its computation load is much less than batch learning, and it is very suitable and compatible with modern computer architectures with multiple GPUs.

2.3 ADNI DATA SET

The data used in this dissertation are obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database. ADNI is a worldwide project that provides reliable clinical data for the research of pathology principle, prevention and treatment of AD. Multiple research groups contribute their findings of the biological markers to the understanding of the progression of AD in the human brain.

For each subject, the T1-weighted MRI was processed by correcting the intensity inhomogeneity followed by skull-stripping and cerebellum removing. In addition, each MRI was segmented into gray matter, white matter and cerebrospinal fluid and was further spatially normalized into a template space. in this dissertation, the gray matter tissue density maps were used. To further reduce the size of image data, the density maps were downsampled to $64 \times 64 \times 64$ voxels. The PET images were also obtained from ADNI.

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

2.4 3-D CNN FOR IMAGING DATA COMPLETION

Convolutional neural networks (CNNs) are a type of deep models that are able to capture highly nonlinear relationships between input and output [2]. In image classification tasks, two types of layers, *i.e.*, convolutional layer and subsampling layer, are usually used in an alternating way. The convolutional layer applies trainable filers to feature maps in the previous layer, while the subsampling layer is used to reduce the resolution of the feature maps.

CNNs have been primarily applied to 2-D images such as visual object recognition [2,3] and segmentation [5]. In [4,6], 2-D CNNs have been extended to segment and restore 3-D images. In [9], 3-D CNNs have been applied to process spatiotemporal video data. Similar to the 2-D case, 3-D CNNs perform nonlinear mapping by computing convolutions with 3-D filters on feature maps.

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

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

where $\sigma(\cdot)$ is the sigmoid function, b_{ij} is the bias for this feature map, m indexes over the set of feature maps in the $(i-1)$th layer connected to the current feature map, P_i, Q_i and R_i are the sizes of the 3-D kernel along three spatial dimensions respectively, w_{ijm}^{pqr} is the (p, q, r)th value of the filter connected to the mth feature map in the previous layer. I demonstrate an example of 3-D convolutional layer in the following Figure **??**.

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

Based on the 3-D convolution described above, a variety of CNN architectures can be devised. In the following, I describe a 3-D CNN architecture, shown in Fig. 1, that I designed for PET image completion on the ADNI data set. The subject pool for training this CNN model consists of the samples having both PET and MRI images.

In this architecture, I start from MRI of $64 \times 64 \times 64$ voxels. To train the network, I randomly select a large number of patches from each 3-D MRI volume. The input patch size is determined by the size of output patch in the output layer, since each convolution operation reduces the size of feature map along each dimension by a factor related to the size of filter. In this dissertation, the size of output patches is set to $3 \times 3 \times 3$. The total number of patches extracted from each volume is $50,000$ so that the entire volume is largely covered.

I then apply 3-D convolution with a filter size of $7 \times 7 \times 7$ on these patches and construct 10 feature maps in the first hidden layer. The second hidden layer is again a 3-D convolution layer with 10 feature maps fully connected to all the feature maps

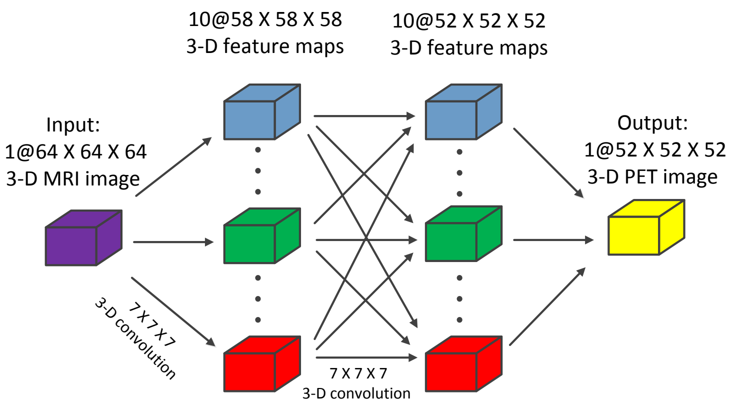


FIG. 1: The architecture of 3-D CNN for imaging data completion used in this dissertation. There are 2 hidden layers between the input MRI volume and the output PET volume. Each of the hidden layers contains 10 3-D feature maps. The filter size of the network is $7 \times 7 \times 7$. The total number of trainable parameters in this network is $37,761$.

in the previous layer. The output layer contains only one feature map, which is the corresponding PET image of the same subject. In addition, the filter size for mapping the feature maps of the last hidden layer to the output was set to 1 to reduce the computational workload. In total, the number of trainable parameters for this architecture is $37,761$. Therefore, the latent nonlinear relationship between the MRI and the PET data was encoded into the large number of parameters in the network. This CNN architecture was selected based on a balance between the representation power and the computational cost of training the network. A network with more layers and feature maps may be able to represent the training data better, but it may also overfit the noise. Additionally, the computational cost of training more complex networks is prohibitive.

In the experiments, I used the 398 subjects with both MRI and PET data to train the network. The weights of this network were updated by error back-propagation using stochastic gradient descent. The learning rate was fixed to 10^{-2} in all the experiments. The network was trained for multiple epochs, where each epoch involves training the network by each example once. in this dissertation, I trained the network for 10 epochs since the performance seems to have converged after 10 epochs and the training was very time-consuming. In particular, I have $398 \times 50,000 = 19.9$ million training patches. Each epoch took approximately 48 hmys if all the 19.9 million patches were used even on a Tesla K20c GPU with 2,496 cores.

2.5 EXPERIMENTAL EVALUATION

2.5.1 EXPERIMENTAL SETUP

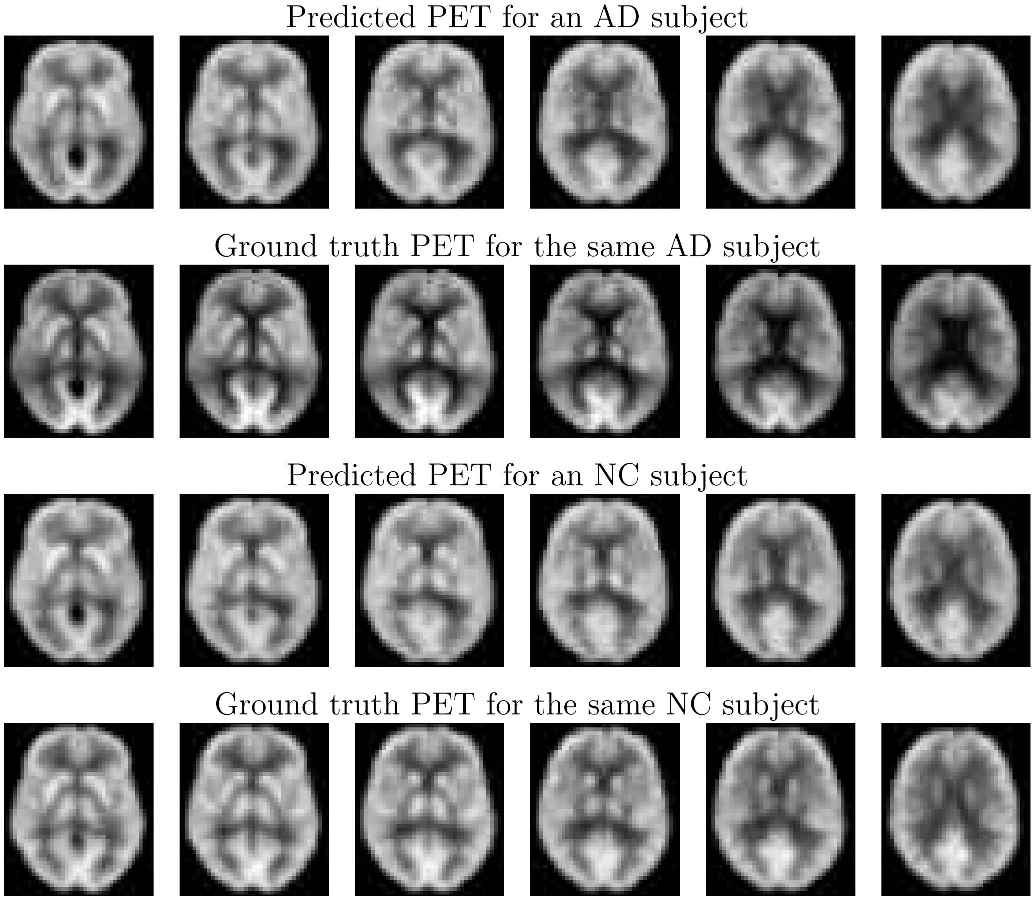
In the experiments, I focus on evaluating my 3-D CNN model for missing PET data completion. To this end, I used several controlled experimental settings to compare the predicted and the true PET image data. I did not employ advanced feature extraction and classification methods to compare the completed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

I compare my deep learning data completion method with two other commonly used missing data estimation methods, namely the K-nearest neighbor (KNN) and the zero methods [10]. In KNN, I first find the top K most similar samples to the target sample without PET images based on MRI image features. Then, I replace the missing values by weighted average of those top K PET features. In the zero method, I first normalize the data set such that it has zero mean and unit standard deviation. Then, I set the missing values to be zero.

The experiments in this dissertation consist of two steps. The first step is to complete the missing PET data using CNN, KNN, or Zero methods. The second step then evaluate the classification performance after completing the missing values. In the experiments, I train the classifiers by randomly selecting 2/3 of the samples and perform the evaluation using the remaining 1/3 as test data in this second step. To obtain robust performance estimates, this random partition is executed 30 times, and the statistics computed over these 30 trials are reported. I perform feature selection by removing voxels that have zero value for all subjects and used L_2-norm regularized logistic regression for classification. Since the classes are unbalanced, I use the area under the ROC curve (AUC) as the performance measure to evaluate different methods in this study. All the experiments are implemented through an open deep learning package called “cuda-convnet” on a GPU K40 with 128Gb memory.

2.5.2 EVALUATION ON SUBJECTS WITH BOTH MRI AND PET

I first evaluate whether the predicted PET data are similar to the true PET data. In the data set used for this dissertation, there are 398 subjects with both MRI and PET images. I randomly sample 1/2 of these 398 subjects for training the 3-D CNN model. Then the model is used to predict the PET images of the remaining 1/2



Predicted PET for an AD subject

Ground truth PET for the same AD subject

Predicted PET for an NC subject

Ground truth PET for the same NC subject

FIG. 2: Comparison of the predicted and the ground truth PET images on two sample subjects.

subjects. Since the test subjects have true PET images, I can compare the true and the predicted PET images both visually and quantitatively.

I first visually examine the predicted PET patterns with the ground truth data slice by slice for each subject. I observe that the predicted PET patterns are very similar to the ground truth data. Figure 14 shows the predicted and the ground truth data slice by slice for two subjects. In figure 14, each row corresponds to the data (either ground truth or predicted) of one subject, and each column corresponds to slice with the same brain position. I observe that the predicted PET patterns are very similar to the ground truth data. This demonstrates that my deep learning based method can accurately predict and complete the missing PET data.

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

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

2.5.3 EVALUATION ON ALL SUBJECTS

In order to evaluate the effectiveness of my proposed method further, I report

TABLE 1: Performance comparison of classification tasks using the true and the predicted PET data. The data set consists of 398 subjects having both MRI and PET images.

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

the prediction performance on all 830 subjects, where 398 subjects have both MRI and PET images, and the remaining 432 subjects have only MRI images. The 3-D CNN and other data completion methods are trained on the 398 subjects, and the trained models are used to complete the PET images of the remaining 432 subjects. The classification performance on all 830 subjects is reported in Table 2. Note that the comparison of classification performance based on true data is not applicable in this experiment, since 432 of the 830 subjects do not have PET images.

I observe that the 3-D CNN model outperforms KNN and Zero methods for all three tasks with three different combinations of PET and MRI image modalities. This again demonstrates that the proposed 3-D CNN data completion method is more effective than prior methods. I also observe the performance is improved when the MRI and PET image features are combined. Overall, these experiments yield

TABLE 2: Performance comparison of classification tasks using the true and the predicted PET images. All 830 subjects were used in this experiments, where subjects with no PET images were completed using three methods.

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

insights on the power of the 3-D CNN model in completing missing neuroimaging data, thereby providing practical guidelines for employing multi-modality data even when some data modalities are missing.

CHAPTER 3

DEEP LEARNING FOR 3D NEURON RECONSTRUCTION

3.1 INTRODUCTION AND RELATED WORK

Morphology of neurons plays a critical role in the function of the brain. Recently, many efforts have been devoted to develop automatic or semi-automatic neuron reconstruction algorithms based on microscopy images. However, this reconstruction task remains very challenging when a 3D microscopy image has low signal-to-noise ratio (SNR) and discontinued segments of neurite patterns. Indeed, microscopy image sets with low SNR are quite common for the nervous systems of different animals [11, 12]. Therefore, an effective and automatic denoising algorithm for these challenging situations would substantially amplify the impact of neuron morphology reconstruction. Currently, most of prior methods for neuronal structure reconstruction on noisy images have used the raw images in an unsupervised way [13–16]. The authors in [11] developed a graph-augmented deformable model to reconstruct the 3D structure of a neuron when the neuron contains a broken structure and/or fuzzy boundary. However, this method still has difficulties on some noisy patterns, *e.g.*, when two parallel tracts are very close and one is brighter than the other. In [11, 17, 18], fast-marching based methods were combined with different pruning

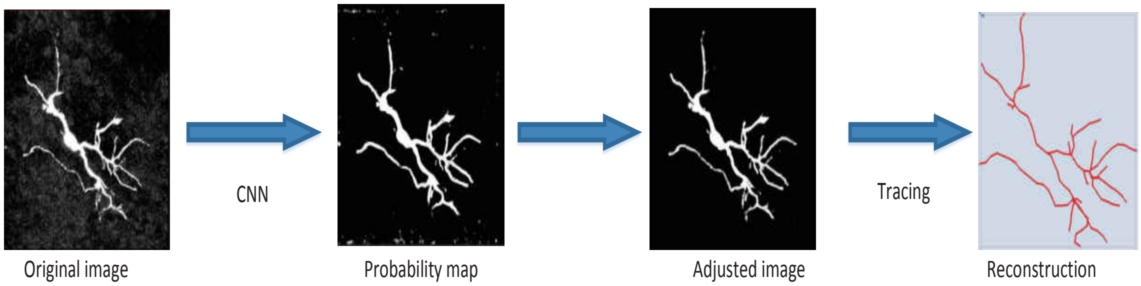


FIG. 3: The pipeline of my method for automated reconstruction of neuronal structures.

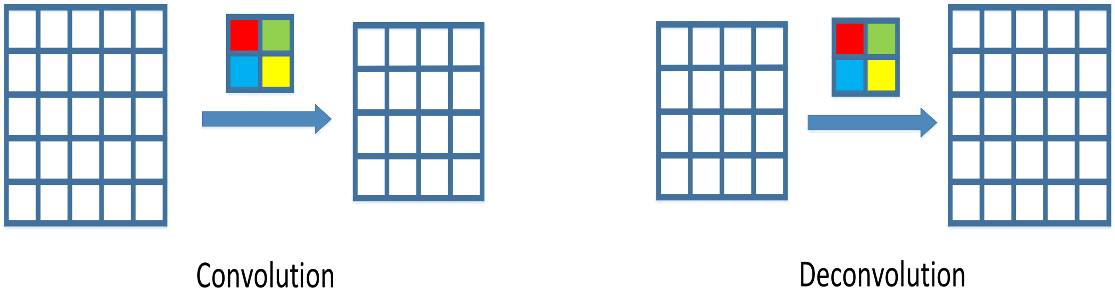


FIG. 4: Comparison of convolution and deconvolution.

techniques to handle 3D noisy images. However, these methods either have low accuracy when the images are anisotropic or are time-consuming when the number of noisy voxels is large. The recently published work [19] developed an automatic tracing framework with training steps to refine the reconstruction results generated by an initial tracing algorithm. One main bottleneck of this method is its relatively high computational complexity.

In this dissertation, I propose to use the convolutional neural networks (CNNs) for

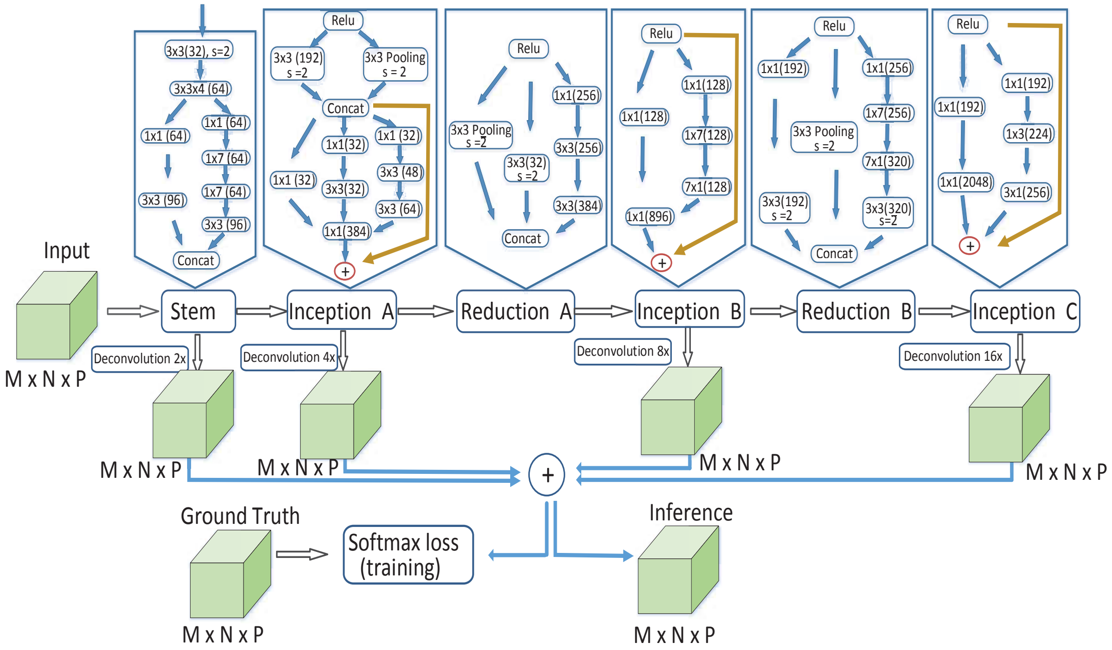


FIG. 5: Detailed architecture of the 3D convolutional neuron network with dense prediction.

improving the reconstruction performance of existing methods on noise-contaminated images over different organisms. CNNs are a type of fully trainable models that learn a hierarchy of features through nonlinear mappings between multiple stacked layers. CNNs have been widely used in a number of applications and achieved state-of-the-art performance on tasks including large-scale image and video recognition [3, 9, 20, 21], digit recognition [5], and object recognition tasks [22]. Recently, many attempts have been made to extend these models to the field of image segmentation, leading to improved performance [4, 6, 23]. One appealing property of CNNs is that the learned

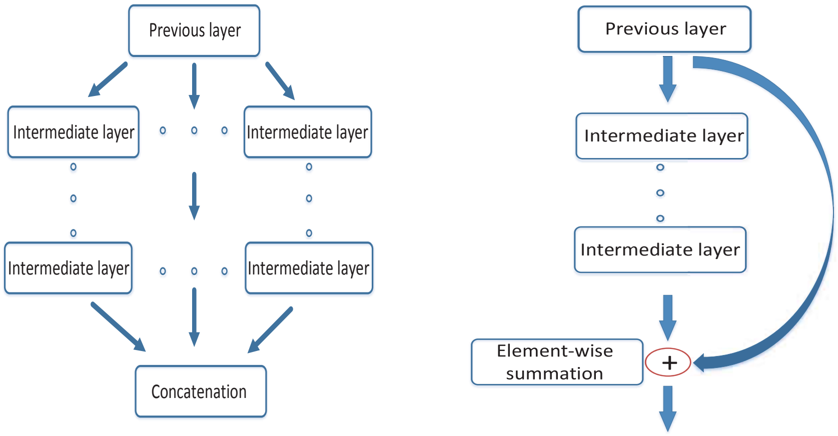


FIG. 6: Illustration of the inception (left) and residual (right) learning.

features through trainable parameters can capture highly nonlinear relationship between inputs and outputs. Therefore, the reconstruction methods based on CNNs can be broadly applied to a variety of different data sets.

Specifically, I built a voxel CNN classifier that predicts the probability of every individual voxel in a given image being a part of neuron or not. CNNs have been primarily applied on 2D images in the prior studies. To effectively incorporate the 3D spatial information in neuronal structures, I propose to perform 3D convolution in the convolutional layers of CNNs so that discriminative features along three spatial dimensions are all captured. The proposed CNN classifier accepts raw voxel intensities as input without any preprocessing and learns highly discriminative features automatically for producing final probability maps. These probability maps are integrated later with the raw intensities to produce the final adjusted image as input for tracing algorithms. In addition, to obtain the predictions of CNNs on test images, the conventional approaches used patch-based predictions. This approach results in significant redundancy in computation, thus making the prediction of large images very time-consuming. To reduce the computational complexity, I apply a stack of deconvolution layers in the CNN architecture that can produce a dense pixel-wise prediction very efficiently. I compare the performance of our approach with that of commonly used tracing methods on a number of challenging 3D neuronal images from different model organisms. Results show that the proposed model significantly outperforms prior methods on neuron tracing from microscopy images.

3.2 BIGNEURON ATLAS

The BigNeuron project [24] provides a large community- oriented neuron morphology database and reconstruction algorithms contributed by many research labs worldwide. There are more than 20,000 volumetric optical microscopy images from a variety of species including fruit fly, fish, turtle, chicken, mouse, rat, and human. The neurons in these images are mainly located in regions such as cortical and subcortical areas, retina, and peripheral nervous system. Each image in BigNeuron has single color channel, and contains a single neuron or disconnected multiple neurons having relatively clear separation in their arborization patterns. Some images have the corresponding reconstructions manually curated and/or proofread to be used as references or “gold standards” for evaluating the automatically produced reconstructions. BigNeuron uses the open-source cross-platform package “Vaa3D” software [25] for facilitating the bench-testing of neuron tracing algorithms developed by worldwide researchers. More than 30 automated neuron-tracing algorithms for neuron quantifications have already been ported to Vaa3D as plug-ins, in addition to a number of

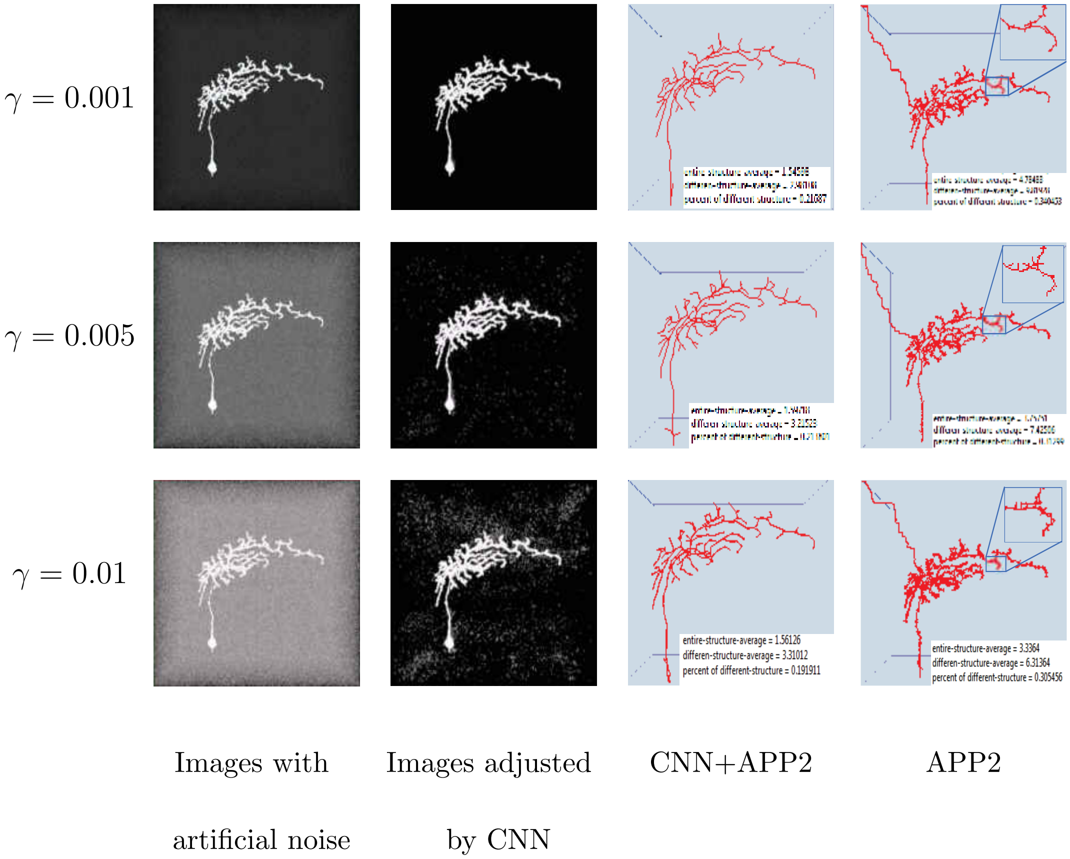

$\gamma = 0.001$
$\gamma = 0.005$
$\gamma = 0.01$
entire-structure-average = 1.54598
percent of different-structure = 0.21087
entire-structure-average = 4.78483
percent of different-structure = 0.340453
differen-structure-average = 3.21523
differen-structure-average = 7.42506
entire-structure-average = 1.56126
differen-structure-average = 3.31012
percent of different-structure = 0.191911
entire-structure-average = 3.3364
differen-structure-average = 6.31364
percent of different-structure = 0.305456
Images with artificial noise
Images adjusted by CNN
CNN+APP2
APP2

FIG. 7: Comparisons of tracing performance on images with artificial noises.

other neuron data analysis tools.

3.3 FAST PREDICTION OF CNNS

One major challenge of using CNNs on neuronal image segmentation is that the volumetric images usually have large sizes and thus could be computational expensive to segment. A common way of generating image segmentations using CNNs is to extract patches from the images and use those patches as inputs to the trained network. The output of each patch is a single label of the center pixel of that patch.

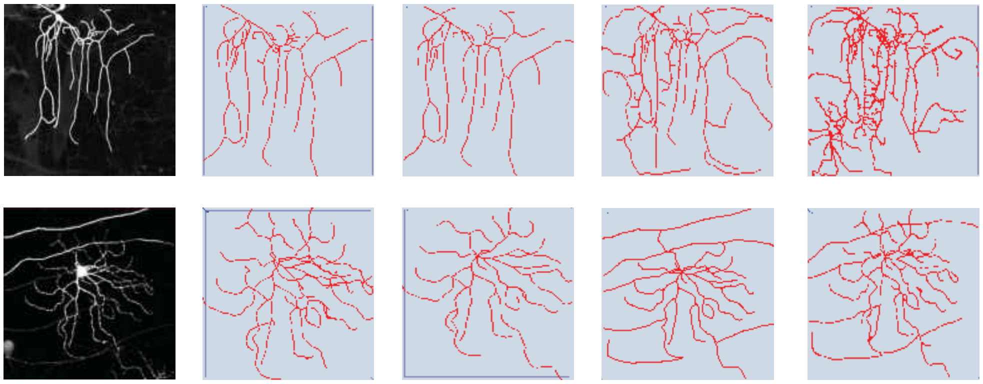


FIG. 8: Comparisons with different neuron tracing methods for two images in BigNeuron database. The first row is for the subject No. 31, and the second row is for the subject No. 52. The first column shows the original images. The second column shows the expert manual reconstruction results. The third column is the reconstructions using APP2 on adjusted images with our CNN model. The fourth column is the results by applying APP2 directly on the original images. The fifth column shows the results by SmartTracing method on the original images.

Such patch-based prediction results in a huge amount of redundant computations. It is thus desirable to design a fast prediction algorithm that can segment the whole image directly without generating patches.

In [26], a fast prediction algorithm was proposed by creating a group of fragments over the feature maps generated from each max-pooling layer. Each fragment contains information independent of other fragments, and the output fragments at the last layer were reorganized to generate the final prediction of the whole image. Although this fast prediction algorithm does not require patch extraction, some

TABLE 3: Quantitative comparisons of tracing performance by APP2 and Smart-Tracing method with and without the adjustment by my CNN model respectively. The results reported in this table are based on the measure "% of different structure".

Sub	CNN+APP2	APP2	CNN+ST	ST	Sub	CNN+APP2	APP2	CNN+ST	ST
3	**0.2075**	0.5347	**0.3153**	0.5339	4	**0.2226**	0.7012	**0.4178**	N/A
5	**0.3472**	0.3483	0.5080	**0.4974**	7	**0.2178**	0.3920	**0.2686**	0.3629
16	**0.2099**	0.2916	0.2488	**0.2204**	17	0.2096	**0.2019**	**0.2968**	0.3134
18	**0.3353**	0.3645	0.3125	**0.3023**	19	0.3010	**0.2748**	0.3276	**0.3036**
20	**0.4588**	0.5684	**0.4092**	0.4130	21	**0.3498**	0.4163	**0.2346**	0.2446
22	**0.3298**	0.3474	0.3188	**0.2901**	25	**0.5443**	0.5727	0.6387	**0.6101**
29	**0.3663**	0.5017	**0.3452**	0.3948	31	**0.1683**	0.3807	**0.2483**	0.4712
35	**0.3622**	0.7447	**0.5120**	N/A	52	**0.2492**	0.4283	**0.2723**	0.3845

duplicate computations still exist when generating different fragments. In [27] the d-regularly sparse kernels were introduced to eliminate the redundant computations in convolutional and pooling layers. Those sparse kernels converted convolution and pooling kernels with various strides into operations with a single stride. This conversion ensured continuous memory access and increased the computational efficiency on GPUs. However, this sparse kernel prediction method is not applicable to architectures in which feature maps are padded before each convolutional layer to retain their sizes.

In this dissertation, I employ a novel network to obtain image predictions efficiently. In particular, I apply deconvolution operations to offset the size reduction caused by convolution and max-pooling operations. Contrary to the convolution

TABLE 4: Tracing performance comparisons using the measure "entire structure average" and "different structure average".

	entire structure average				**different structure average**			
	CNN+APP2	APP2	CNN+ST	ST	CNN+APP2	APP2	CNN+ST	ST
Sub. 3	**1.4667**	19.1914	**1.9345**	13.253	**3.7271**	24.3327	**3.8273**	17.6276
Sub. 4	**1.4608**	20.4576	**2.2307**	N/A	**2.9812**	24.4982	**3.6339**	N/A
Sub. 5	**2.9562**	3.4425	2.9451	**2.8185**	**5.7559**	6.6293	4.1684	**4.0008**
Sub. 7	**1.7715**	4.1832	**2.2249**	4.1815	**3.1334**	7.6208	**4.8516**	7.8795
Sub. 16	**1.7643**	2.7168	**2.1222**	2.3085	**4.2014**	6.9710	**4.2785**	5.2957
Sub. 17	**1.5133**	1.5218	**1.9180**	1.9883	**2.7030**	2.8417	**2.8988**	3.0038
Sub. 18	**1.9768**	2.1165	2.5506	**2.4962**	**3.1877**	3.3327	4.8305	**4.7152**
Sub. 19	1.7361	**1.6803**	2.0086	**1.9669**	**2.7909**	2.7932	3.0154	**2.9459**
Sub. 20	**2.2600**	4.4414	**2.5101**	3.0851	**3.2338**	6.3721	**3.9653**	5.1144
Sub. 21	**1.9406**	2.9079	**1.7357**	1.9998	**3.0792**	5.3018	**2.9771**	3.7801
Sub. 22	**2.0178**	2.1073	**2.0318**	2.0717	**3.3857**	3.5341	**3.2598**	3.5386
Sub. 25	**15.1363**	25.2285	**16.9098**	18.8010	**18.9222**	29.3581	**20.0443**	22.1727
Sub. 29	**2.3182**	4.3457	**2.5062**	4.2284	**3.7772**	7.1298	**4.4599**	8.1930
Sub. 31	**1.5740**	13.4657	**1.8931**	10.0305	**5.3058**	23.2479	**4.0464**	13.5121
Sub. 35	**3.0737**	31.0310	**3.4451**	N/A	**6.6046**	32.0973	**5.2681**	N/A
Sub. 52	**4.3104**	17.6192	**3.3240**	12.0861	**17.4785**	37.7065	**9.8060**	25.2226

operation that connects multiple input activations using a convolution kernel to produce a single output, as shown in Figure 4, the deconvolution associates a single input with multiple outputs using a deconvolution transformation. One significant advantage of using deconvolutional layers is that they allow the output feature map to have the same size as the input image by using carefully selected deconvolution kernel and stride sizes. Specifically, I use multiple deconvolution operations at different intermediate layers to make the sizes of feature maps at those layers the same. The deconvolved same-size feature maps are then combined to form a multi-scale representation of the model input. Through such design, the proposed network is capable of generating an end-to-end mapping between inputs and outputs and leads to dense pixel-wise prediction over images without any computational redundancy.

3.4 THE PROPOSED DEEP MODEL

Another major challenge of using CNNs for neuronal image segmentation is the large diversity of the images in the BigNeuron database. First, the complicated structures of neurons require the network to be able to learn features from multiple scales. For example, the recognition of neuron skeleton needs large filters, since neurons can be very thin yet projecting to long distance in 3D space. On the other hand, the fine neuronic structures like branches or bifurcations needs small filters to capture the local information in a small neighborhood. Second, images in the BigNeuron datasets are contributed by different research groups and acquired using different imaging techniques on various species such as fruit fly, fish, etc. This results in a large amount of differences of neuron morphology among the images. Hence, the network is expected to extract the essential common characteristics of neurons over those various samples. Third, the qualities of images in BigNeuron are dramatically different, which introduced an extra difficulty for the model training. For example, the signal levels of key neurite in some images might be strong and thus easy to detect, but these signals might be weak in other images because of multiple factors such as inappropriate exposure, noises from non-neuronal cells, or limitations of experimental devices.

To overcome the above-mentioned difficulties, I propose a novel 3D CNN for reconstructing neurons accurately. Prior studies have demonstrated the superior performance of CNNs on classifications in natural images. However, in those applications, 2D convolutions are applied to mimic the visual information processing. When applied to the neuron reconstruction problems, it is desirable to employ 3D convolutions since neurons are naturally in 3D space. Architectures with 3D convolutions have been successfully used in video analysis [9, 28–30], in which the video data were viewed as a 3D volume with time acts as the third dimension. The authors in [31] used 3D filters to build nonlinear mappings between different image modalities, but its architecture was too shallow and inefficient for our neuron tracing task. In order to tackle the large variance of neuronal structures in 3D space, I propose a fully convolutional network (FCN) with 3D convolutions to localize the neurons in the BigNeuron database. FCNs are variants of CNNs by replacing the fully connected layers with convolutional layers. Such model allows the network to operate on inputs of any size and produce outputs of corresponding spatial dimensions. Fully convolutional layers together with deconvolutional layers ensure the feasibility of end-to-end training and testing. Furthermore, in order to improve the performance of networks, the strategy of stacking more layers is widely used in literature. Such strategy has two major drawbacks. One is deeper network typically means a larger number of parameters, which increases the risk of overfitting. The other drawback is the dramatically increased computational complexity. In this dissertation, I propose to use two powerful deep learning techniques introduced in the following to overcome these limitations.

3.4.1 INCEPTION AND RESIDUAL LEARNING

One typical way for overcoming the above mentioned two limitations of deeper networks is to introduce small kernels instead of large kernels. However, large kernels are still necessary for capturing information of large regions in the input. In order to deal with this dilemma, inception networks [32, 33] used multiple kernel sizes and max pooling in parallel in each stage, and then aggregated their outputs for the next stage, as shown in Figure 6. In each inception module, prior to expensive convolutions with large filter sizes, convolutions with small kernels are inserted to reduce the dimensionality such as the number of feature maps. An advantage of such design is that it allows for increasing the number of units at each stage significantly without an uncontrolled blow-up in computational complexity at later stages. Meanwhile, the use of multiple filter sizes ensures that the hidden information can be processed at various scales simultaneously.

The residual learning technique [34] was initially proposed to solve the degradation problem in which the training accuracy decreases when using too many stacked layers. Formally, we use $H(x)$ to denote the desired nonlinear mapping between the input feature map x and the output feature map after applying stacked layers. In residual learning, the stacked convolutional layers were considered to only fit $H(x)$'s nonlinear residual $F(x) := H(x) - x$. In other word, the mapping $H(x)$ can be reformulated as $F(x) + x$ and realized by adding a shortcut identity mapping connection between the input feature map and the output map, as shown in Figure 6. Such design ensures that residual networks can achieve more accurate results using deeper models but the training will be still very fast. This is because previous layers are reused by subsequent layers through shortcut connections, which makes each layer learns less than a plain network but utilized more information.

In this dissertation, I use residual learning for segmenting the neuron morphology from optical microscopy images. In addition to its excellent performance in dealing with the degradation problem, another motivation of using residual learning is the limited number of samples for training. Although the number of images in the BigNeuron database is large, only a small number of them have manual reconstructions by experts. For segmentation tasks over 2D natural images, transfer learning could be used to borrow weights from pre-trained models on other similar data sets. However, transfer learning may not work in the segmentation of 3D neuron images because of the absence of pre-trained models for transferring. This lack of training samples restricted the depth of the model since deep models with limited samples may overfit. Due to the appealing property of residual learning, I employ shortcut connections between both convolutional and deconvolutional layers to construct the hierarchical representation of neurons.

3.4.2 THE DETAILED ARCHITECTURE

I provide the detailed configuration of the proposed deep network in Figure 5.

The whole network contains 6 modules, and each of them used multiple paths for realizing inception learning. In particular, I use 3 modules (inception A, B, C in Figure 5) mainly for building the nonlinear relationship between input and output, and an additional 3 modules mainly for reducing the sizes of feature maps (reduction A, B, C in Figure 5). For each module in the architecture, convolutional layers are denoted by kernels sizes and number of feature maps in the brackets. Except those layers used stride size of 2, which are indicated by "s=2", the stride sizes in other layers are all 1. The filter sizes in third dimension is selected as 1 in most layers, and I ignore to notify that for simplicity. The orange arrows stand for the shortcuts of residual learning. Multiple deconvolutional layers are used for up-sampling the intermediate feature maps to have the same size as the input. All outputs of deconvolutional layers are summed together to form a multi-scale representation of inputs. In those 3 inception modules, residual learning is applied for improving the performance and also reducing the computation complexity. The network is 48-layer deep when counting only layers with parameters. The overall number of layers (including scaling and batch normalization) used for the construction of the network is over 200.

All the convolutional layers, including those inside the inception modules, and deconvolutional layers use rectified linear activation. Considering the limited number of foreground neuron voxels in the original 3D images, I randomly sample the background voxels for each training subject in order to control the numbers of positive and negative samples. Intuitively, voxels around the neuron boundary are more difficult to classify correctly than other distant voxels. Therefore, most of the background samples are drawn around the neuron boundary. Also the ratio between the numbers of foreground and background voxels are kept within an acceptable level. Note that the numbers of voxels in foreground and background classes are not balanced. If the contribution of each class to the loss is treated equally during the training, the obtained network will mainly capture the discriminative features of the majority class which is the background class in this case. Therefore, in the loss layer, I apply different weights to the foreground and background classes to overcome this imbalance problem.

3.5 IMAGE ADJUSTMENT

For each microscopy neuron image, I can generate its probability map P having the same size as the image. Each element of P indicates the probability of the corresponding voxel as neuron, known as foreground probability. A natural way to use the probability map is to apply the tracing algorithm directly to detect the neuronal structure. In this dissertation, I combine the original image and the probability map together to get a combined representation for suppressing the noise signals effectively. Specifically, for an input image $I(x)$ where x is a voxel, I screen the probability map by a threshold δ chosen empirically to identify the foreground voxels. If the foreground probability $P(x)$ of the voxel x is less than δ, I set the intensity $I(x)$ to zero, otherwise I keep its original intensity value unchanged. Thus, an intermediate intensity image $\tilde{I}(x)$ is defined as

$$\tilde{I}(x) = \begin{cases} I(x) & \text{if} \quad P(x) > \delta, \\ 0 & \text{if} \quad P(x) \leq \delta. \end{cases}$$

In order to further use the probability map, I construct a new probability image $\tilde{P}(x)$ by multiplying the probability value $P(x)$ with 255 and rounding the decimal value to the closest integer. The final adjusted image $F(x)$ by my method is defined as

$$F(x) = \alpha\tilde{I}(x) + (1 - \alpha)\tilde{P}(x)$$

where α is a weight to control the contributions of the original intensity and the probability map. Then the tracing algorithm will be applied on the adjusted image to trace the final neuronal reconstruction. The detailed pipeline of the proposed method is illustrated in Fig. 12.

3.6 EXPERIMENTAL EVALUATION

3.6.1 EXPERIMENTAL SETUP

In this dissertation, I select 68 subjects from the BigNeuron database. These subjects are from a variety of species, and each subject has the corresponding expert manual reconstruction as ground truth for training and evaluation. Out of these 68 3D images, more than half of them contain clearly visible noises in the images. I randomly sample 3/4 of these 68 subjects for training the proposed models. Then the model is evaluated on the remaining subjects to compute the neuron tracing performance. The training and testing split is performed in a stratified way in terms of data source to avoid the scenario that all subjects from some research labs are all either in training or test sets.

During the training phase, I trim off the background margins of the training images according to the ground truth for saving computational resources. In order to improve model performance, I use data augmentation to enlarge the training data set. The data augmentation includes transformations of the original images with rotation and flipping along different dimensions. During the test phase, I empirically select the probability threshold value δ in 2 to be 0.02. The coefficient α in Eqn. 3.5 is tuned through line-search over test images and the value of 0.85 is used. I apply three distance scores to measure the difference between a particular reconstruction and the ground truth. These scores were defined in [11] and are known as "entire structure average", "different structure average" and "percentage of different structures". Specifically, for each node in manual reconstruction, I calculate the minimal spatial distance between this node and all nodes in the reconstructed nodes generated by computational methods. The entire structure average is obtained by averaging all these reciprocal minimal spatial distances; the different structure average only sums those distances that are larger than 2 voxels since the difference of two nodes that is less than 2 voxels are hardly distinguished visually. The percentage of different structures captures the percentage of pairs of nodes whose reciprocal minimal spatial distances are more than 2 voxels. For all of these three measurements, larger values indicate higher discrepancy between the tracing results and the manual reconstruction.

3.6.2 PERFORMANCE ON IMAGES WITH ARTIFICIAL NOISE

A major advantage of the proposed CNN model is that I can obtain accurate segmentations of neurons. Such segmentations can enhance the signal of neurons from a noisy image, which improves the neuron tracing applied in the following. In order to demonstrate the robustness of our CNN models with respect to different levels of noises, I add Gaussian white noise of mean 0 and different variance γ to the original image of Subject 17. I then use each image contaminated with artificial noise as the input to my CNN model, and generate the corresponding adjusted image as described in Section 3.5. For both contaminated images and adjusted images by CNN models, I apply the APP as the tracing method to reconstruct the neuron structures. The visualization results for three values of $\gamma = 0.001$, 0.005 and 0.01 are given in Figure 7. The first column shows the images of Subject No. 21 with different levels of added Gaussian white noises. The variance values are 0.001, 0.005, 0.01, respectively, from top to bottom. The second column shows the corresponding adjusted images after applying our CNN model to remove the noise voxels. The third column shows the reconstruction results by the APP method on the adjusted images. The fourth column shows the reconstruction results by the APP method on the contaminated images. The up-right boxes in the fourth column show the parts of the corresponding images. The distances between the tracing result and the ground truth is also give in each image in the third and fourth columns.

I can observe that after applying our proposed segmentation model, the numbers of noise pixels in the adjusted images are significantly decreased. The reconstructions on the adjusted images are visually close to the ground truth by human experts. In contrast, the reconstructions on the contaminated images without adjustments contain many unnecessary spurs because of the noise signals. In addition, I compute the distances between the reconstructions and the ground truth. I observe that for different levels of noise, the performance on the adjusted images is consistently better than that on the contaminated images. This shows that the trained CNN models could be potentially used to improve the accuracy of neuron tracing, especially if the microscopy images are contaminated by noises.

3.6.3 PERFORMANCE ON REAL IMAGES

In order to further demonstrate the advantage of our CNN models on denoising neuron microscopy images, I use two real images with visible noises contributed by different research labs. In addition, I compare my method with the SmartTracing method [19] which is another learning-based tracing framework. I use APP2 as the tracing method applied on the original and adjusted images, since SmartTracing is also based on APP2. The reconstruction results of different methods are given in Figure 8.

I observe that the tracing results on adjusted images are visually more similar to the ground truth than those on the original images. The probability maps of our CNN model captured those bright noise voxels, which facilitated the reconstructions applied subsequently. In contrast, many redundant branches and bifurcations are still wrongly kept as neuron segments by APP2 on the original images. Although SmartTracing is a also learning based method, it used local features with complicated transformations to train the model. In contrast, CNNs employ stacked layers with different filter sizes to extract multi-scale information of objects in the image. The advantage of the proposed model over SmartTracing can be observed in Figure 8.

3.6.4 COMPARISON WITH OTHER METHODS

In order to evaluate the proposed method quantitatively, I compare the tracing performance of APP2 and SmartTracing with and without adjustment by the proposed CNN model respectively. The results on all 17 test subjects are reported in Table 3, 4. I observe that, for both APP2 and SmartTracing, their performance on adjusted images is better than that based on the original images for all three measurements. Specifically, APP2 with CNN adjustment outperforms APP2 on the original images for all subjects with respect to the measure “entire structure average”. Even for the other two measures, there are only one or two subjects on which APP2 without CNN yield slightly better performance. Similar performance gains by the proposed CNN models are also observed on the results using SmartTracing. In addition, I observe that for those images with a large amount of noise such as Subject 3, 4, 31, and 52, the tracing results after using our CNN models are significantly better than those on the original images. These results further demonstrate that the proposed 3D CNN method is very effective in improving neuronal reconstruction on noisy images.

CHAPTER 4

DEEP LEARNING FOR PROTEIN STRUCTURE DETECTION

4.1 INTRODUCTION AND RELATED WORK

Proteins perform most of the work of living cells with unique and stable three-dimensional (3D) structures. The detection of the folded structures of proteins is critical for understanding the working mechanisms of action in large macromolecular complexes such as viruses, cellular organelles and membranes, etc. In the past decades, a number of studies have been conducted to develop efficient methods for determining the structures of proteins which are categorized to four levels. Most of prior progresses in this field are based on experimental techniques such as X-ray crystallography and nuclear magnetic resonance (NMR). Cryo-electron microscopy (cryo-EM) is an experimental technique with increasing popularity to study the structures of protein complexes. Through cryo-EM, a number of large molecular complexes, such as ribosome and viruses, have been resolved to near atomic resolutions.

However, detecting the secondary structures of proteins automatically and accurately at medium resolutions (5-10Å) is still a challenging task. The major difficulty is from the fact that the spatial shape patterns of secondary structure elements (SSEs) at medium resolutions are hardly distinguished with their closely located neighbours. The most common secondary structure elements (SSEs) are α-helices,

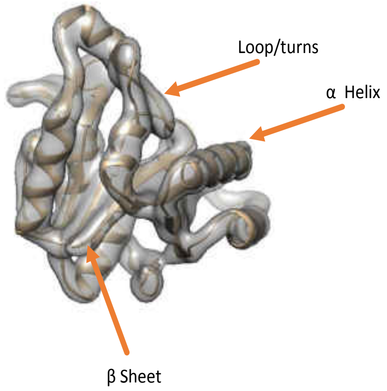
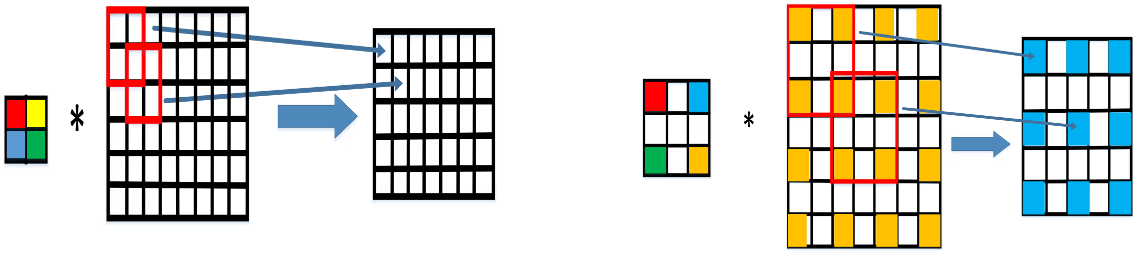


FIG. 9: Illustration of different SSEs in an example protein including α-helix, β-sheet and turns/loops. The backbone of a protein structure is shown as a ribbon, and the surface view of the corresponding density image is superimposed as well.

β-sheets, and turns/loops. An example showing their shapes is given in Figure 9. In general, the long α-helices and large β-sheets can be detected accurately because of their easily recognized spatial shapes. However, short α-helices appear to be similar to turns/loops in the density maps at the medium resolution and they are hard to distinguish. For instance, a β-sheet with two strands can be easily confused with an α-helix. The shape and structure of turns/loops are also very similar to those of the neighbour α-helices. Therefore, it is highly demanded for efficient computational methods that can accurately and automatically identify SSEs from EM images at medium resolutions containing high similarities.

Currently, most prior methods for detecting SSEs at medium resolutions are based on image-processing techniques. These methods search for cylinder-like regions for α-helices and plane-like regions for β-sheets. In particular, the methods in [35, 36] detected the α-helices computationally from a density map at sub-nano resolution.



Original convolution

Dilated convolution

FIG. 10: Illustration of the original convolution and the dilated convolution.

They used a five-dimensional cross-correlation search of a three-dimensional density map followed by a series of integrated feature-detection steps. The authors of [37,38] studied the unique morphological features of SSEs, and then used the structural similarities based on these features for locating α-helices and β-sheets. In addition, De novo modeling [39] was designed to derive the backbone from the specific density maps of low resolutions. In general, these existing methods have two common drawbacks. The first one is they need user interventions such as the careful adjustment of various parameters. The second drawback is that they did not fully explore the existing data to assist in detecting SSEs of new samples. The database size of cryoEM images for protein backbones is rapidly increasing. It is thus desirable to utilize the existing data in a more efficient way for improving the performance of detecting protein structures. Recently, learning based methods with few user interventions are attracting more research attentions in detecting protein SSEs. The studies in [40] used a nested K nearest neighbors classifiers for improving the α-helices detection. However, it used different proteins in the same cryoEM map to form the training

TABLE 5: Accuracy of identified Cα atoms from the simulated maps.

PDB ID	1ajw	1ajz	1al7	1cv1	1dai	1eny	1wab	2aw0	2itg	3lck	Average
Total	145	282	350	162	219	268	212	72	160	270	
Hlx	5	124	159	123	84	126	96	22	66	107	
tp HlX	5	120	155	119	82	124	96	22	66	98	
m HlX	0	4	4	4	2	2	0	0	0	9	
fp Hlx	26	23	35	21	24	60	57	11	31	25	
Sht	63	37	46	14	47	66	24	25	21	30	
tp Sht	60	32	41	12	45	46	24	25	21	29	
m Sht	3	5	5	2	2	20	0	0	0	1	
fp Sht	61	18	14	18	16	18	22	31	49	24	
Spe.Hlx	81.43	85.44	81.68	46.15	82.22	57.75	50.86	78	67.02	84.66	71.52
Sen.Hlx	100	96.77	97.48	96.75	97.62	98.41	100	100	100	91.59	97.86
Spe.Sht	25.61	92.65	95.39	87.84	90.70	91.09	88.30	34.04	64.75	90	76.04
Sen.Sht	95.24	86.49	89.13	85.71	95.74	69.70	100	100	100	96.67	91.87

and testing data. The authors in [41] developed a machine learning framework based on support vector machine (SVM) method to automatically identify α-helices and β-sheets with usage of other existing volumetric maps. However, the sample features they used are locally hand-crafted features which are not representative enough for the essential characteristics of protein structures.

In this dissertation, I propose to use the convolutional neural networks (CNNs) for detecting secondary structures in protein EM images. CNNs are a type of fully trainable models that learn a hierarchy of features through nonlinear mappings between multiple stacked layers. CNNs have been widely used in a number of image related applications and achieved state-of-the-art performance on tasks including large-scale image and video recognition [3,9,20,21], digit recognition [5], and object recognition tasks [22]. Recently, many attempts have been made to extend these models to the field of image segmentation, leading to improved performance [4,6,23]. One appealing property of CNNs is that the learned features through trainable parameters can capture highly nonlinear relationship between inputs and outputs. Therefore, it is natural to employ CNNs for obtaining high representative features from EM images to improve the performance of detecting protein SSEs.

Specifically, I built a voxel CNN classifier that predicts the probabilities of every individual voxel in a given EM image with respect to different kinds of SSEs. To effectively incorporate the 3D spatial information in protein structures, I propose to perform 3D convolution in the convolutional layers of CNNs so that discriminative features along three spatial dimensions are all captured. The proposed CNN classifier accepts voxel EM densities as input and learns highly discriminative features automatically for producing intermediate label prediction. These intermediate label predictions are then integrated with post processing steps for detecting the final secondary structures. In addition, the conventional approaches used patch-based predictions to obtain the outputs of CNNs on test images, which is very time-consuming for large images. To reduce the computational complexity, I apply a stack of deconvolution layers in the CNN architecture that can produce a dense pixel-wise prediction very efficiently. I compare the performance of my approach with that of commonly used learning based methods on a number of challenging 3D cryoEM density maps. Results show that the proposed model significantly outperforms prior methods on detecting secondary structures of proteins from volumetric images.

4.2 THE PROPOSED DEEP MODEL

One major challenge of using CNNs for cryo-EM segmentation is the large image diversity in the database. The complicated structures of proteins require the network be able to learn features from multiple scales. For example, the recognition of α-helices needs large filters, since an α-helix usually extends long in 3D space. On the other hand, detecting β-sheet structures needs small filters to capture the local information in a short and flat neighborhood. To overcome the above-mentioned difficulties, I propose a novel 3D CNN for recognizing the protein SSEs from cryoEM images accurately. To be specific, I design a fully convolutional network (FCN) with 3D convolutions to localize the SSEs in the images. The proposed fully convolutional layers integrated with deconvolutional layers ensure the feasibility of end-to-end training and testing. The detailed introduction of FCNs and deconvolutional layers can be found in the previous chapter.

Furthermore, I also propose to use inception learning and residual learning to improve the performance of networks. An inception network employs different kernel sizes and also max pooling as multiple pathes in each stage, and then collect their outputs for the next stage, which allows for increasing the number of units at each stage significantly without sharply increasing the computational complexity at later stages. The residual learning is designed to simulate the desired nonlinear mapping between the input and output of some stage by adding a shortcut identity mapping connection. The residual networks can achieve more accurate results using deeper models with a high speed still. The description of these two techniques are omitted here since the detailed introduction of them are also provided in the previous chapter.

4.2.1 DILATED CONVOLUTION

Many image related applications such as semantic segmentations problems required the developed model could keep the local pixel-level accuracy such as precise detection of edges, and also utilize the knowledge from the wider global context. To this end, researchers have developed various techniques in deep learning field for acquiring the multi-scale representation of the input. Besides the inception learning and residual learning introduced in the above two sections, the convolution with a dilated filter has also been studied and shown excellent performance in many computer vision applications.

The convolution with a dilated filter is an extension of the original convolution. Its significant property is that the dilated convolutions support exponentially expanding receptive fields without losing resolution or coverage. Therefore, a neural network with it could capture information from different scales without increasing the number of parameters too much. In particular, the formula of the original convolution over the 1-D input signal f with the kernel k is defined as follows,

$$(k * f)_t = \sum_{\tau=-\infty}^{\infty} k_\tau \cdot f_{t-\tau},$$

where t is the variable of f. Instead, the convolution with a dilated filter factor l between f and k is defined as:

$$(k *_l f)_t = \sum_{\tau=-\infty}^{\infty} k_\tau \cdot f_{t-l\tau}$$

TABLE 6: Quantitative comparisons of detection performance by our deep learning model (DL) and support vector machine method (SVM) for α-helix and β-sheet respectively. The averaged F1 scores over these two structures are also reported.

Protein ID	**F1 Helix**		**F1 Sheet**		**F1 average**	
	DL	SVM	DL	SVM	DL	SVM
1ajw	27.78	8.26	65.22	72.41	**46.50**	40.34
1ajz	89.89	66.67	73.56	34.26	**81.73**	50.47
1al7	88.83	66.67	81.19	34.85	**85.01**	50.76
1cv1	90.49	86.83	54.55	21.05	**72.52**	53.94
1dai	86.32	58.87	83.33	49.47	**84.83**	54.17
1eny	80.00	66.84	70.77	50.38	**75.39**	58.61
1wab	77.11	66.67	68.57	30.38	**72.84**	48.53
2aw0	80.00	47.83	61.73	56.18	**70.87**	52.01
2itg	80.98	67.35	46.15	27.63	**63.57**	47.49
3lck	85.22	64.42	69.88	35.29	**77.55**	49.86
Average	78.66	60.04	67.50	41.19	**73.08**	50.62

In the dilated convolution, the kernel only touches the signal at every lth entry. This formula applies to a 1-D signal, but it can be straightforwardly extended to higher dimensional convolutions. The detailed difference between these two kinds of convolution is illustrated in Figure 10.

Recently, dilated convolutions have been employed for semantic segmentation on natural images. The authors in [42] analyzed filter dilation and done preliminary experiments for comparisons with other developed tricks. The authors of [43] used dilated convolutions to simplify the architecture of [42]. A new convolutional network architecture that systematically uses dilated convolutions is proposed in [44]

TABLE 7: Quantitative comparisons of detected secondary structures by CNN and SSEHunter [36].

Protein ID	**CNN**			**SSEHunter**		
	Hx < 5Å	Hx 5-8Å	Sh=2Str.	Hx < 5Å	Hx 5-8Å	Sh=2Str.
1ajw	0/0	1/1	0/0	0/0	1/1	0/0
1ajz	0/1	3/3	0/1	0/1	3/3	0/1
1al7	2/3	4/4	1/2	1/3	4/4	0/2
1cv1	0/1	2/2	0/0	1/1	0/2	0/0
1dai	1/2	2/2	2/2	2/2	2/2	0/2
1eny	0/0	1/1	0/0	0/0	1/1	0/0
1wab	3/3	0/0	0/0	1/3	0/0	0/0
2aw0	0/0	0/0	0/0	0/0	0/0	0/0
2itg	0/0	1/1	0/0	0/0	1/1	0/0
3lck	1/4	2/2	1/1	1/4	0/2	1/1
Total	7/14	16/16	4/6	6/14	12/16	1/6

for multi-scale context aggregation. In [44], the spatial pooling layers were replaced with convolutions with increased filter dilated sizes. In this Chapter, I propose to integrate the dilated convolution with inception and residual learning to build an efficient neural network for identifying protein SSEs from cryoEM images. To be specific, I used the dilated convolution at the inception modules of Figure 5 respectively. Before the residual learning is implemented, I used the dilation convolution with filter size 3 for obtaining a larger size of receptive fields.

4.2.2 THE DETAILED ARCHITECTURE

I provide the detailed configuration of our proposed deep networks. In order to obtain competitive performance, I average the outcomes of two different models to obtain the final probability map about the α-helix and β-sheet voxels. The first CNN model I use is similar to the one introduced in Figure 5. The whole network also contains 6 modules, and each of them used multiple paths for realizing inception learning. There are also 3 modules (inception A, B, C in Figure 5) mainly for building the nonlinear relationship between input and output, and an additional 3 modules mainly for reducing the sizes of feature maps (reduction A, B, C in Figure 5). The major difference between the architecture of this chapter with the one demonstrated in Figure 5) is the usage of dilated convolutions. To be specific, for each inception modules, I introduce one extra dilated convolution layer with kernel size of 3 and dilation factor size of 1 on the top of those concatenation layers before the residual learning is applied. The detailed figure is omitted here since the architecture is very similar with Figure 5). The other architecture I use is the one proposed in the paper [45]. Through the ensemble of these two models, the final performance of identification on SSEs is improved since the biases are averaged and the variances are reduced.

4.3 EXPERIMENTAL EVALUATION

4.3.1 EXPERIMENTS SETUP

In this dissertation, I select 25 simulated cryoEM images for training and testing. These cryoEM images are acquired by an open software package called “chimera” [46] to perform semiautomated single-particle reconstructions from transmission electron

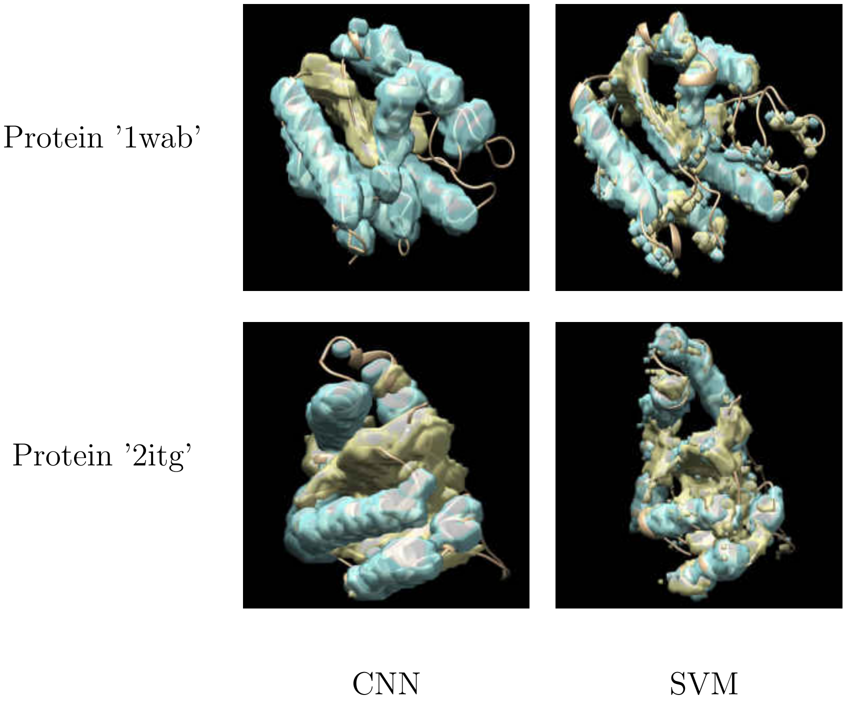


FIG. 11: Comparisons with different identification methods for two testing cryoEM images. The first row is for the protein ’1wab’, and the second row is for the protein ’2itg’. The first column shows the result by proposed CNN method. The second column shows the results by SVM method. The results for both two methods are generated without any postprocessing steps.

micrographs. In particular, I generate the training and testing simulated maps to 8Å resolution using the program command “pdb2mrc” of “EMAN43” with a sampling size of 1Å/pixel. I randomly sample 15 of these 25 subjects for training the proposed models. Then the models are evaluated on the remaining subjects to compute the SSE detection performance.

The performance is firstly evaluated by calculating the number of identified secondary structures and also the number of Cα atoms that falls in the neighborhood of the secondary structures. To be specific, a helix is identified if its length is within one turn difference from the length of the helix in the PDB structure. A β-sheet is identified if the identified β-sheet voxels visually overlay on the β-sheet of the PDB structure. To present a more quantitative estimation about the size of the identified helices and β-sheets, I estimate the number of Cα atoms that are close to the identified helix voxels and β-sheet voxels. The detailed selections of parameters for closeness are similar to those given in [41].

I also use the F1 score to quantitatively measure the segmentation accuracy on detecting helix and sheet Cα atoms. Let A and B denotes the number of positive elements in the binary segmentation labels generated manually and computationally, respectively, about one protein SSE class on pixels for certain cryoEM image. The F1 score is defined as

$$F1 = \frac{2|A \cap B|}{|A| + |B|},$$

where $|A|$ denotes the number of positive elements in the binary segmentation A, and $|A \cap B|$ is the number of shared positive elements by A and B. The $F1$ score lies in [0,1], and a higher value indicates a higher detection accuracy.

During the training phase, I trim off the background margins of the training images according to the ground truth for saving computational resources. In order to improve model performance, I use data augmentation to enlarge the training data set. The data augmentation includes transformations of the original images with rotation and flipping along different dimensions. During the testing phase, I use the fast prediction technique introduced in previous chapter for obtaining the output probability values over SSEs.

4.3.2 PERFORMANCE ON SIMULATED EM DENSITY MAPS

In order to demonstrate the effectiveness of the proposed method, I firstly report the specificity and sensitivity based on the detected helix and sheet Cα atoms. In this chapter, I consider a Cα atom as an identified helix Cα if it is in the neighbourhood of an identified helix voxel with the radius of 2.5Å. Similarly, an identified sheet Cα should be within the neighbourhood with the radius of 3Åof an identified sheet voxel. The detailed numbers of identified Cα atoms for all the testing cryoEM density maps are given in Table 5. The first column shows the total number of Cα atoms in the protein. The total numbers of Cα atoms in helices and β-sheets are listed respectively in second and sixth columns. The numbers of true positive, missed, and false positive Cα atoms for helices and β-sheets are give in third to fifth, and seventh to ninth columns respectively. The specificity and sensitivity of helix and sheet detection are reported in the last four columns, respectively.

I can see that the sensitivity and the specificity of helix identification can reach 71.52% and 97.86%, respectively. The sensitivity and specificity for β-sheet identification is 76.04% and 91.87%, respectively.

In order to provide a comprehensive and quantitative evaluation of the proposed method on detecting protein SSEs, I also report the identification performance on all 10 testing cryoEM images. The performance of our proposed method and another learning-based method using SVM is reported in Table 6 using F1 scores.

I can see that our method outperformed the existing method for both α-helix and β-sheet detection. Specifically, CNN could achieve F1 score as 78.66% for α-helix and 67.5% for β-sheet on average over 10 testing subjects, yielding an overall value 73.08%. In contrast, SVM method achieves F1 score as 60.04% for α-helix and 41.19% for β-sheet, yielding an overall value 50.62%.

In addition to quantitatively demonstrating the advantage of the proposed CNN method, I visually exam the identification results of α-helix and β-sheet for two testing samples in Figure 11. The ground truth three-dimensional structural morphologies are shown with color ribbons. The first column presents the results of CNN method and the second columns are for the SVM method. It can be seen that identified SSEs by CNN method are quite similar to the ground truth while SVM generates more false positives.

In order to further compare results by different methods, the numbers of detected secondary structures by CNN model and the the method called “SSEHunter” proposed in [36] are also presented in Table 7. I can see that, through CNN method, 8 out of the 14 extremely short α-helices of size less than 5 amino acids are identified. For detecting α-helices of size more than 5 amino acids, the CNN model is also comparable with SSEhunter. For the Compared with SSEhunter's results, the proposed CNN model is able to detect 4 out of 6 β-sheets with 2 strands. These results further show that the proposed method is more effective than other method.

CHAPTER 5

IDENTIFICATION OF CELL-TYPE-SPECIFIC GENES

5.1 INTRODUCTION AND RELATED WORK

Although all cells in the brain are genetically identical, they can develop into different cell-types that are distinct in morphology, connectivity, and function. For example, the mammalian brain contains an enormous number of neuronal and glial cells. The neuronal cells are responsible for information communication and processing, while the glial cells are traditionally considered to provide supportive functions. Cell-type diversity is resulted from the different sets of molecules that cells of each type contain. This is in turn due to the differential expression and regulation of genes in the genome. Thus, analysis of gene expression patterns provides an informative way of studying cellular diversity [47, 48]. In these studies, it has been commonly observed that some genes are specifically expressed in certain cell-types. These genes serve as cell-type markers and might define cell-type-specific transcriptional programs [49, 50]. A complete catalogue of the cell-type-specific genes would be valuable in elucidating the relationship between gene expression patterns, connectivity, brain regions, and cell-types [51–55]. Currently, both experimental and computational approaches have been used to study cell-type-specific gene expression patterns. Experimental methods involve in separating cells of different types from heterogeneous tissues and measuring gene expression levels in the separated

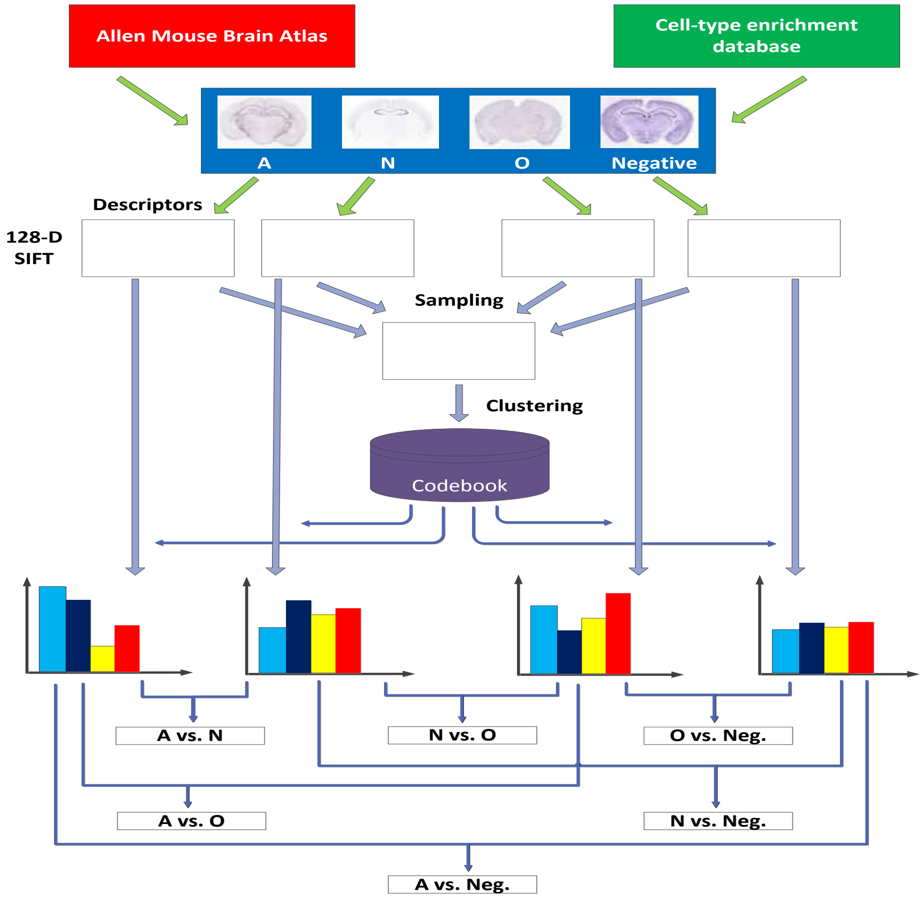


FIG. 12: The pipeline of my methods for automated identification of cell-type-specific genes.

tissues using microarrays. Along this line, multiple techniques have been developed for tissue processing; they, however, suffer from different limitations [49]. As an alternative approach, current computational methods identify cell-type-specific genes by comparing their expression profiles captured by either microarrays [56–58] or *in situ* hybridization (ISH) voxel-level data [59]. These approaches either lack the fine spatial resolution or the high-order expression characteristics that are needed for resolving cell-type-specificity.

In this dissertation, I aim at identifying cell-type-specific genes by mining and analyzing the high-resolution ISH expression pattern images directly. I apply invariant image feature descriptors to compute high-order expression characteristics from ISH images. These descriptors were computed on dense and overlapping local patches, leading to millions of descriptors from each ISH image section. They collectively capture the local gene expression information, and the spatial information is implicitly encoded into the overlapping patches. To obtain image-level representations, I first cluster these descriptors to obtain the visual words that represent the dominant local expression patterns. I then compute a bag-of-words representation for each ISH image by constructing a histogram based on the visual words. This representation counts the frequency of each visual word occurring in each ISH image, forming a high-level representation of an ISH image. I employ regularized learning methods for discriminating genes specifically expressed in different major brain cell-types, namely, neurons, astrocytes, and oligodendrocytes [47]. The proposed method can also identify the visual words that are most distinct between different brain cell-types [60]. To obtain a robust estimation of the most discriminative visual words, I employ stability selection to construct an ensemble model. The pipeline of my proposed methods is depicted in Figure 12.

The experimental results show that the high-level representations computed directly from cellular-resolution ISH images are predictive of cell-type-specificity of genes in major brain cell types. I use the area under the receiver operating characteristic curve (AUC) as the performance measure [61, 62]. I achieve AUC values of approximately 87% in five out of the six tasks when the threshold value for fold enrichment is set to 20, a recommended value based on experimental data [47]. The results also show that the image-based invariant representations for ISH images generally yield better performance than voxel-based features in discriminating genes enriched in different brain cell types. The average AUC value given by my image-based approach on data sets with >1.5 enrichment fold is approximately 75% while an average AUC value of 65% is achieved by voxel-based features. Visualization of highly-ranked features show that they corresponded to locations containing the most discriminative features among brain cell-types. I also compare the performance of different tasks to investigate the intrinsic relationship between various brain cell-types. My results show that the relative performance differences among various brain cell-types are generally consistent with my current knowledge on cell-type functions.

5.2 ALLEN MOUSE BRAIN ATLAS

The Allen Mouse Brain Atlas provides genome-wide, three-dimensional, high-resolution *in situ* hybridization (ISH) gene expression images for approximately 20,000 genes in the sagittal section for the 56-day old male mice [63]. In addition, coronal sections at a higher resolution are available for a set of about 4,000 genes showing restricted expression patterns. For each experiment, a set of high-resolution, two-dimensional image series are generated. These image slices are subsequently processed by an informatics data processing pipeline to generate grid-level voxel data in the Allen Reference Atlas space [64]. The output of the pipeline is quantified expression values at a grid voxel level [65, 66]. The voxel-level data have been used to identify cell-type-specific genes based on correlation search [59]. Note that the selection of coronal genes was biased toward genes enriched in cortical and/or hippocampal regions [67].

5.3 ISH IMAGE FEATURE EXTRACTION AND CONSTRUCTION

To fully exploit the cellular-resolution ISH images and extract high-order information for classification, I compute features from the original ISH images directly. The ISH images I use are taken from different mouse brains. Thus, the shape and size of the brain and various anatomical structures might vary from image to image. Additionally, tissue processing and image acquisition might also introduce distortions on the images. To account for these image-level variations, I employ the scale-invariant feature transform (SIFT) descriptor to capture expression patterns on local patches of ISH images [68, 69]. This approach can produce robust representations that are invariant to various distortions on the images. To compute SIFT features, an image is first convolved with a sequence of Gaussian filters of different scales to produce difference-of-Gaussian (DOG) images. Stable key-point locations are then detected from these DOG images. A set of orientation histograms on 4×4 neighborhoods at each location are subsequently computed, and each histogram contains 8 spatial bins recording the pixel gradients in 8 orientations.

In many of the current image classification systems, key-point extractors are typically not used [70, 71]. Instead, SIFT features are commonly applied on regularly spaced grid on the images, leading to densely populated SIFT descriptors. Following [72, 73] I also apply dense SIFT features on the ISH images [74]. This generated approximately 1 million SIFT feature vectors from each ISH image section [72]. In this dissertation, I use the most medial slice of each sagittal section image series. For the coronal section image series, I use the slice with the median Section ID that corresponds to the middle location between the most posterior section showing the cerebellum and hindbrain and the most anterior section showing the olfactory bulb. The use of more slices would incur high computational cost. In addition, it has been shown in [72] that performance may not be improved when more slices were used. In the Allen Mouse Brain Atlas, a detection algorithm is applied to each ISH image to create a mask identifying pixels in the ISH image that correspond with gene expression. Thus foreground pixels are considered to correspond with gene expression while background pixels are not [63]. Only the SIFT descriptors computed from the foreground pixels are used in this dissertation.

In order to derive an image-level representation for cell-type-specific gene classification, I employ the bag-of-words method to construct ISH image representations [75–77].

The bag-of-words representation was originally used in text mining and is now commonly employed in image and video analysis problems in computer vision. In this approach, invariant features are first extracted from local regions on images or videos, and a visual codebook is constructed by applying a clustering algorithm on a subset of the features, where the cluster centers are considered as “visual words” in the codebook. Each feature in an image is then quantized to the closest word in the codebook, and an entire image is represented as a global histogram counting the number of occurrences of each word in the codebook. The size of the resulting histogram is equal to the number of words in the codebook and hence the number of clusters obtained from the clustering algorithm. The codebook is usually constructed by applying the flat k-means clustering algorithm or other hierarchical algorithms. This approach is derived from the bag-of-words models in text document categorization, and is shown to be robust to distortions in images. One potential drawback of this approach is that the spatial information conveyed in the original images is not represented explicitly. This, however, can be partially compensated by sampling dense and redundant features from the images. The bag-of-words representation for images is shown to yield competitive performance on object recognition and retrieval problems after some postprocessing procedures such as normalization or thresholding. The the basic idea behind the bag-of-words approach is illustrated in Figure 13. To construct a visual codebook, I randomly sample the non-zero descriptors of every image to obtain a descriptor pool of size 100,000. In some of the classification tasks, the numbers of images in the two classes differ significantly. To take this situation into account, I equalize the number of descriptors chosen from both classes. That is, approximately half of the sampled descriptors are from each of the two classes. The descriptors from each class are equally distributed among all images in that class.

I apply the k-means algorithm to cluster the SIFT descriptors in this pool. S-ince the k-means algorithm depends on the initialization, I repeat the algorithm

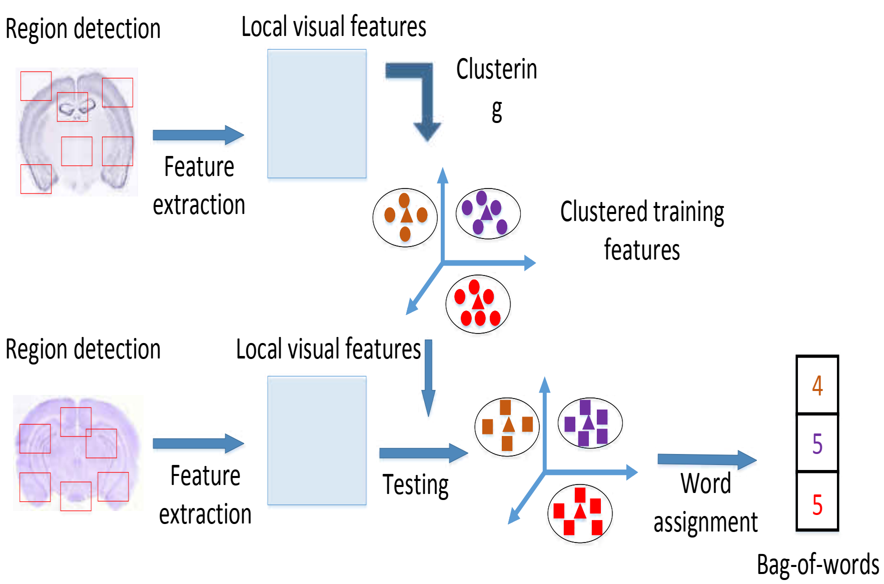


FIG. 13: Illustration of the bag-of-words scheme.

three times with random initializations and use the one with the smallest summed within-cluster distance. The cluster centers are considered as “visual words” in the codebook. I then represent an entire image as a global histogram counting the number of occurrences of each visual word in the codebook. The size of the resulting histogram is equal to the number of words in the codebook, which is also the number of clusters used in the clustering algorithm.

Formally, let $\mathbf{c}_1, \ldots, \mathbf{c}_m \in \mathbb{R}^d$ be the m cluster centers (visual words), and let $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{R}^d$ be the n SIFT features extracted from an image, where $d = 128$ for SIFT. Then the bag-of-words representation $\mathbf{x}$ is m-dimensional, and the k-th component x_k of $\mathbf{x}$ is computed as

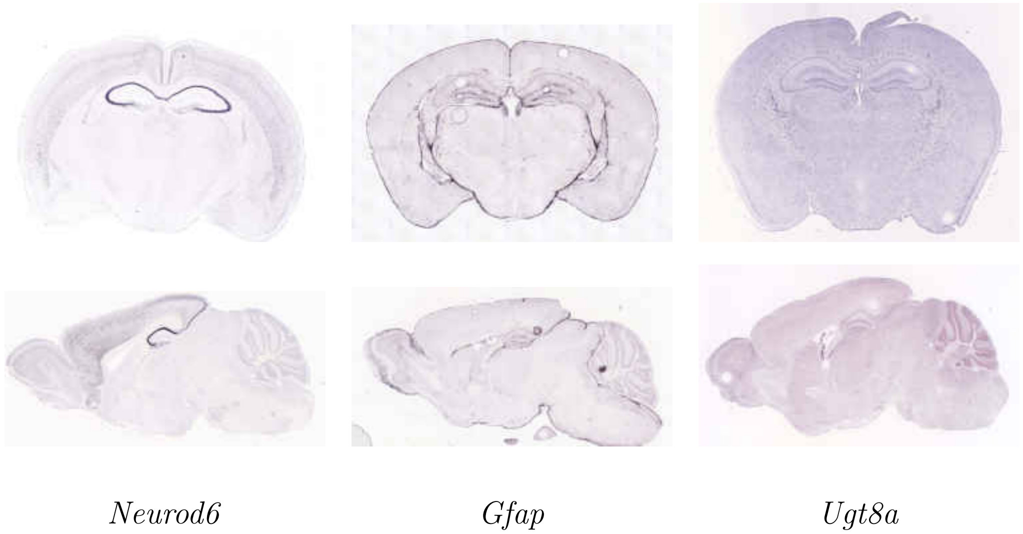
$$x_k = \sum_{i=1}^{n} \delta(k, \arg\min_j ||\mathbf{v}_i - \mathbf{c}_j||),$$

where $\delta(a, b) = 1$ if $a = b$, and 0 otherwise, and $|| \cdot ||$ denotes the vector ℓ_2-norm.

To capture the spatial expression patterns at different scales, I construct four separate codebooks for images with four different resolutions. I then quantize each image using multiple bags of visual words, one for each resolution. The representations for different resolutions are then concatenated to form a single representation for the image. Following [72], I fix the number of clusters to be 500 in the reported results. To account for the zero descriptors, I introduce an extra dimension in the histogram to record the number of zero descriptors for each image at each resolution. Eventually, an ISH image is represented by a high-level feature vector $\mathbf{x} \in \mathbb{R}^p$, where $p = (500 + 1) \times 4 = 2004$. Note that the bag-of-words representation has been successfully applied to represent biological images in the past [72, 78]. In addition, the local binary pattern (LBP) features have been used in [79] to identify genes expressed in cerebellar layers. I compare the LBP features with the bag-of-words features and observe that the later performs better for the problem studied in this dissertation.

5.4 CELL-TYPE-SPECIFIC GENE CLASSIFICATION

I identify the cell-type specificity of genes by classifying the high-level image feature representations constructed above. To achieve this, I need a data set of genes with the corresponding cell-type specificity for training and evaluating my methods. In [47], the fluorescent-activated cell sorting technique was used to isolate and purify the astrocytes, neurons, and oligodendrocytes from the developing mouse forebrain. The expression levels of over 20,000 genes in these cell types were then measured using microarrays, providing a quantitative, genome-wide characterization of the



Neurod6 *Gfap* *Ugt8a*

FIG. 14: Sample ISH images for genes *Neurod6*, *Gfap*, and *Ugt8a* that are enriched in neurons, astrocytes, and oligodendrocytes, respectively. Selected images from approximately the same location were shown for coronal (top) and sagittal (bottom) sections.

gene expression levels in different brain cell types. By comparing the expression levels of genes across these major brain cell types, three lists of genes enriched in astrocytes, neurons, and oligodendrocytes, respectively, were generated and ranked based on the folds of enrichment. The expression patterns of some example genes enriched in each of the three cell-types are displayed in Figure 14. Note that the data in [47] were obtained from the mouse forebrain, instead of the whole brain.

In this dissertation, I train and evaluate my methods based on the genes enriched in astrocytes, neurons, and oligodendrocytes [47]. For each gene studied in [47], I check the availability of ISH images from the Allen Mouse Brain Atlas. By doing this, I obtain a database consisting of 6,660 ISH image series representing 2,872 genes in total. Note that each gene in this database could be associated with more than one cell type, though this does not happen very often.

Each gene in this database is associated with one class label, which is either one of the three cell-types or a negative class label when it does not belong to any of the three classes. To discriminate genes with different class labels, I design six classification tasks by constructing different positive and negative data samples. In the first three tasks, I use genes enriched in one specific cell-type as positive examples and the negative samples consist of genes with negative class labels. For the other three tasks, I design classification tasks to discriminate genes enriched in different brain cell-types. Results show that classification of genes enriched in different brain cell-types yielded insights on the cell-type relationships. The statistics on the numbers of images and genes for these six tasks when the threshold for fold enrichment is 1.5 are given in Table 8. The pipeline of the proposed methods is depicted in Figure 12.

Given a set of training samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^p$ denotes the input feature vector, and $y_i \in \{-1, 1\}$ denotes the corresponding output label. In the problem considered in this dissertation, $\mathbf{x}_i$ represents the bag-of-words feature vector, and y_i encodes the cell-type enrichment information of the corresponding gene. I employ the following regularized formulation for classification:

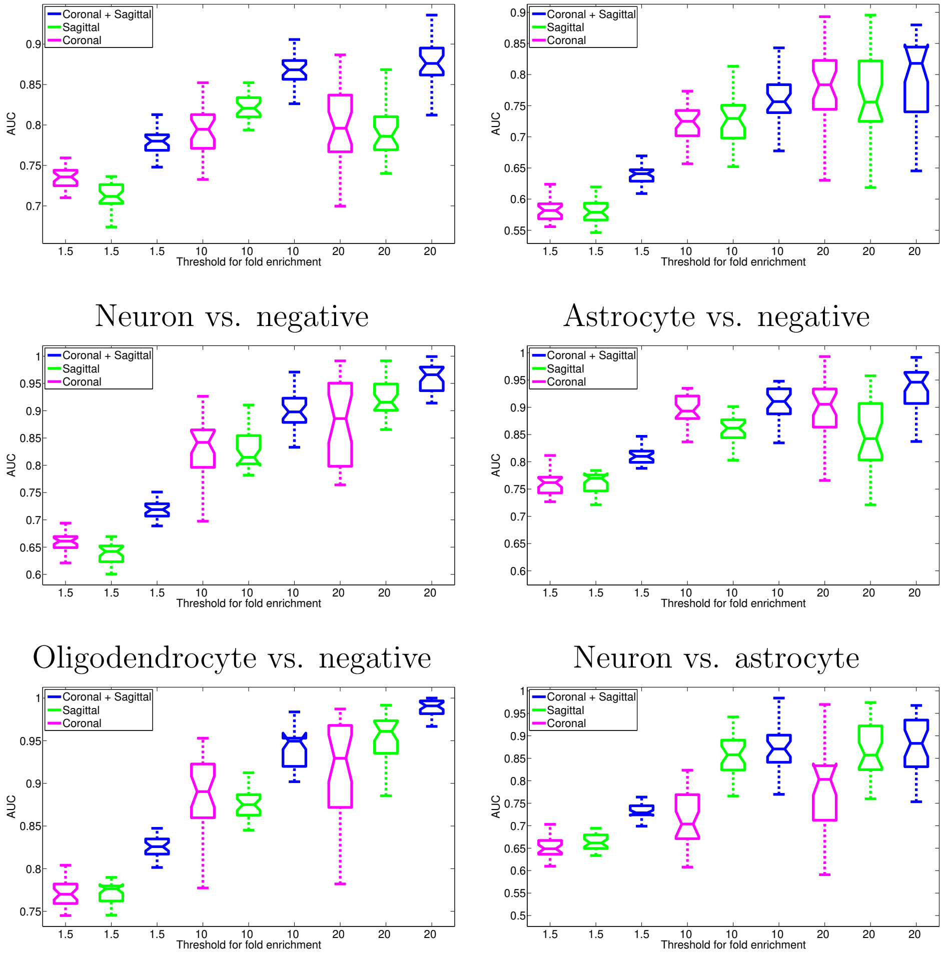
$$\min_{\mathbf{w}} \sum_{i=1}^{n} L(\mathbf{w}^T\mathbf{x}_i + b, y_i) + \lambda\Omega(\mathbf{w}), \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^p$ and $b \in \mathbb{R}$ denote the model weight vector and bias term, respectively, $\Omega(\mathbf{w})$ denotes the regularization term, and λ is the regularization parameter.

TABLE 8: Statistics on the numbers of images and genes for each of the six tasks with different thresholds for fold enrichment.

Folds	Tasks	Number of genes	Number of images
1.5	A vs. Neg.	711 vs. 939	775 vs. 981
	N vs. Neg.	775 vs. 939	844 vs. 981
	O vs. Neg.	541 vs. 939	577 vs. 981
	O vs. A	501 vs. 671	532 vs. 730
	A vs. N	690 vs. 754	754 vs. 823
	N vs. O	753 vs. 519	819 vs. 552
10	A vs. Neg.	72 vs. 939	80 vs. 981
	N vs. Neg.	178 vs. 939	209 vs. 981
	O vs. Neg.	47 vs. 939	50 vs. 981
	O vs. A	47 vs. 72	50 vs. 80
	A vs. N	72 vs. 178	80 vs. 209
	N vs. O	178 vs. 47	209 vs. 50
20	A vs. Neg.	26 vs. 939	31 vs. 981
	N vs. Neg.	67 vs. 939	78 vs. 981
	O vs. Neg.	17 vs. 939	18 vs. 981
	O vs. A	17 vs. 26	18 vs. 31
	A vs. N	26 vs. 67	31 vs. 78
	N vs. O	67 vs. 17	78 vs. 18

In this dissertation, I employ the logistic regression loss function as this loss yielded competitive performance in classification tasks [80, 81]. The ℓ_2-norm regularization $\Omega(\mathbf{w}) = \|\mathbf{w}\|_2$ is used when making predictions [82]. Additionally, I am interested in identifying the most important image features that contributed to the classification performance. This can be achieved by employing the ℓ_1-norm regularization $\Omega(\mathbf{w}) = \|\mathbf{w}\|_1$, which drives some entries of $\mathbf{w}$ to zero, leading to feature



Neuron vs. negative

Astrocyte vs. negative

Oligodendrocyte vs. negative

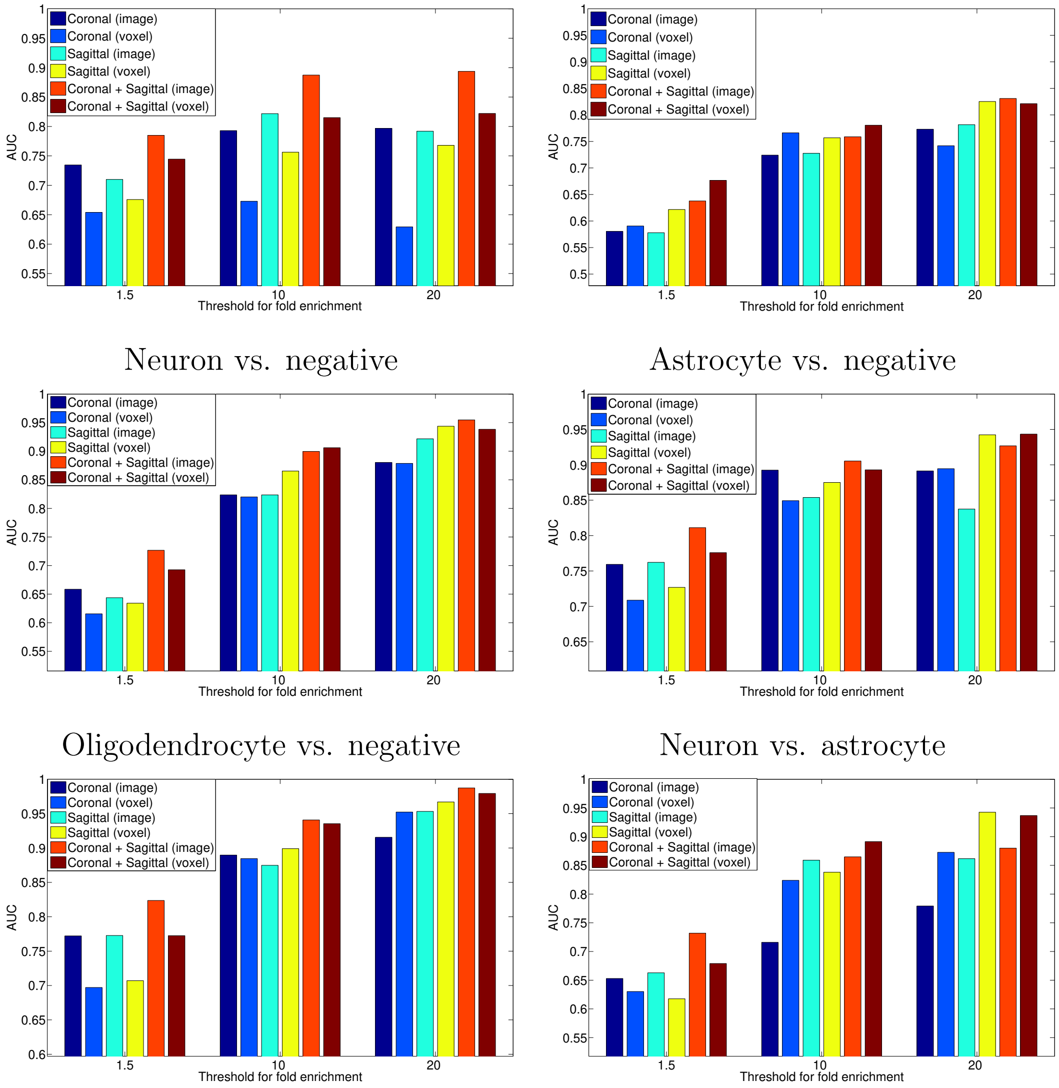
Neuron vs. astrocyte

Neuron vs. oligodendrocyte

Astrocyte vs. oligodendrocyte

FIG. 15: Box plots of the classification performance achieved on the six tasks.

selection [83–88]. To make the ℓ_1-norm based feature selection robust and stable, I employ an ensemble learning technique known as stability selection [89, 90]. In this technique, a set of λ values are selected, and data sets of size $\lfloor n/2 \rfloor$ are repeatedly sampled, without replacement, from the original data of size n. For each sampled data set, a set of models, corresponding to different λ values, are trained. Then the selection probability for each feature under a particular λ value is computed as the



Neuron vs. negative

Astrocyte vs. negative

Oligodendrocyte vs. negative

Neuron vs. astrocyte

Neuron vs. oligodendrocyte

Astrocyte vs. oligodendrocyte

FIG. 16: Comparison of classification performance achieved by my image features and that by the voxel features used in prior work. The performance on the six tasks are compared, and nine data sets are used for each task. For each task on a specific data set, the entire data set is randomly partitioned so that 2/3 of the data is in the training set and the rest 1/3 is in the test set. A total of 30 random partitions are generated, and the average performance is reported. The numbers of genes used for different tasks are given in Table 8.

TABLE 9: Statistical test results in comparing my image-based method with voxel-based method. I employed two-sided Wilcoxon signed rank tests on the AUC values produced by 30 random trials, and the *p*-values were reported. I also performed the one-sided statistical test to compare the mean of image-based multiple trials with that of voxel-based method. The bold values indicate tasks on which image-based method outperforms voxel-based method significantly.

Folds	Sections	A vs. Neg.	N vs. Neg.	O vs. Neg.	O vs. A	A vs. N	N vs. O
1.5	Coronal	0.0822	**1.7E-06**	**4.7E-06**	**0.0036**	**1.7E-06**	**1.7E-06**
	Sagittal	1.7E-06	**8.5E-06**	0.1306	**2.9E-6**	**1.7E-06**	**2.1E-06**
	Cor.+Sag.	3.5E-06	**1.2E-05**	**0.0017**	**1.7E-06**	**2.6E-06**	**1.7E-06**
10	Coronal	6.6E-04	**1.7E-06**	0.9263	9.3E-06	**8.7E-05**	0.7343
	Sagittal	0.0558	**1.7E-06**	5.5E-4	0.0916	0.0180	0.0052
	Cor.+Sag.	0.0387	**1.1E-05**	0.5038	0.1086	0.3389	0.4908
20	Coronal	**0.0612**	**1.9E-06**	0.9590	0.0001	0.7188	0.0100
	Sagittal	0.0387	**0.0026**	0.0157	2.7E-5	5.7E-6	0.0614
	Cor.+Sag.	0.6435	**9.7E-05**	**0.0114**	4.0E-4	0.3359	**0.0349**

relative frequency that this feature is selected among the multiple random samples. Finally, the maximum selection probability across the λ values is computed and used to rank the features.

5.5 EXPERIMENTAL EVALUATION

I formulate the prediction of cell-type-specific genes as a set of six binary-class classification tasks. The prediction is performed by using ℓ_2-norm regularized logistic regression [91]. I also employ the ℓ_1-norm regularized logistic regression [85] and stability selection for image feature ranking. For each prediction task, I use the area under the ROC curve (AUC) as the performance measure [61, 62]. I randomly partition the entire data set for each task into training and test set so that 2/3 of the data are in the training set, and the remaining 1/3 are in the test set. To obtain robust performance estimation, this random partition is performed 30 times, and the statistics computed over these 30 trials are reported.

In [47], genes with >1.5-fold enrichment are reported for each of the astrocyte, neuron, and oligodendrocyte cell types. It is also stated in [47] that genes enriched with >20-fold should be considered as cell-type-specific based on the enrichment levels of well-established cell type markers. In [50] genes with >10-fold enrichment were considered as cell-type-specific genes. I thus generate multiple data sets by using 1.5, 10, and 20 as cutoff enrichment levels for each of the six tasks. The numbers of genes and images in each task are summarized in Table 8.

In the Allen Mouse Brain Atlas, ISH images are provided in both the sagittal and the coronal sections, and I use only those genes with both coronal and sagittal data. I extract SIFT features and constructed high-level representations for the coronal and the sagittal images separately. Since images from different sections might capture different and complementary information, I also concatenate the coronal and sagittal representations in the classification tasks. To ensure that all features have the same dimensionality, the codebook size is reduced to 250 so that the concatenated features have the same dimensionality as the features constructed from only coronal and sagittal images. I also use the same set of genes for the coronal and the sagittal images so that the results are directly comparable.

5.5.1 PERFORMANCE OF GENE IDENTIFICATION

I report the predictive performance achieved by the proposed methods on different data sets in Figure 15 using box plots. Each plot corresponds to one of the six tasks, and nine different data sets are generated by using different thresholds for the fold enrichment and different image sections (coronal, sagittal, and coronal + sagittal). For each task, the entire data set is randomly partitioned so that 2/3 of the data is in the training set and the rest 1/3 is in the test set. A total of 30 random partitions are generated. The central mark represents the median, the edges of the box denote the 25th and 75th percentiles. The whiskers extend to the minimum and maximum values not considered outliers, and outliers are plotted individually. The numbers of genes used for different tasks are given in Table 8.It can be observed from the results that the predictive performance is generally higher on data sets with larger enrichment fold cutoff values. This result is consistent with the fact that genes with large enrichment folds tend to have more cell-type-specificity and thus were easier to identify by my computational methods. In addition, I can observe that combination of the coronal and the sagittal images invariably yields higher performance than either the coronal or the sagittal images individually, suggesting that different sectional images capture complementary information.

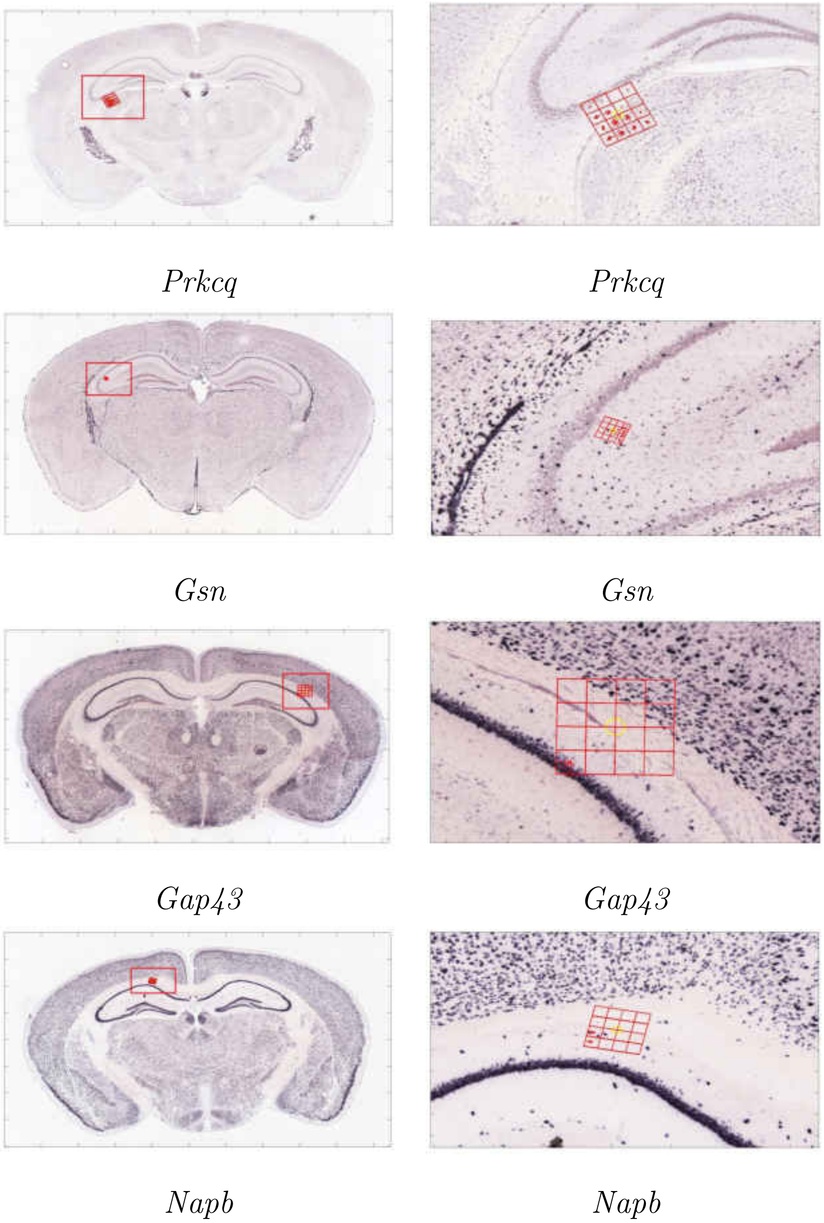
I also consider the performance achieved by the combination of the coronal and sagittal images, as these data sets yield the best performance. When the enrichment fold cutoff value is set to 1.5, the performance on five out of the six tasks is higher than 0.7. When the cutoff value is increased to 10, the performance on five out of the six tasks reaches 0.85. When the cutoff value is further increased to 20, the performance on five out of the six tasks becomes higher than 0.87. Note that a comparative study in [47] showed that genes enriched with >20-fold should be considered as cell-type-specific. At this level, my proposed methods can achieve high predictive performance. These results demonstrate that my image-based predictive methods are able to identify cell-type-specific genes in major brain cell types.

5.5.2 COMPARISON WITH OTHER METHODS

The initial attempt to identify cell-type-specific genes from the ISH data used the grid-level voxel data generated from the registered ISH images [59]. In particular, [59] used well-established cell-type marker genes as queries to identify genes enriched in the same cell-type. This was achieved by computing the correlations of all other genes with these marker genes based on the voxel-level expression grid data. A high correlation value is considered as a high probability of enriching in the same cell-type. I compare the voxel-based features and my image-based features in identifying cell-type-specific genes in a discriminative learning framework.

Specifically, I compare the performance of methods using two different types of data, namely the voxel-level expression energy values and the invariant feature representations computed directly from the ISH images. To this end, I use the grid-level expression energy values as features and build discriminative classifiers as I do with my image-based features. That is, I employ the same set of protocols but replace my image-based features with the voxel-based features where all annotated voxels are used. The results for all six tasks are given in Figure 16. To evaluate the statistical significance of the performance differences, I perform two-sided Wilcoxon signed rank tests on the AUC values produced by 30 random trials, and the p-values are reported in Table 9.

I can observe from these results that, in the neuron vs. negative classification task, my image-based method significantly outperforms the voxel-based method on all nine data sets. In contrast, these two methods yield similar performance in classifying astrocyte vs. negative images, and most of the performance differences on this task are statistically not significant. Note that from the results reported below in Figure 18 I can see that the astrocyte vs. negative task gives the lowest overall performance among all six tasks. Thus, it seems that astrocyte-enriched genes are intrinsically difficult to identify, regardless of the feature representations used. For the other four tasks, I observe that my image-based method outperform voxel-based method consistently and significantly on data sets with >1.5 enrichment fold. For instance, the average AUC value given by my image-based approach is approximately 0.75 while the average AUC value achieved by voxel-based features is approximately 0.65. The performance on other data sets is generally similar, and the differences are mostly not significant. These results demonstrate that my image-based invariant representations are generally better than voxel-based features in discriminating genes enriched in different brain cell types. In addition, the differences are particularly apparent for genes with low cell-type-specificity.



Prkcq *Prkcq*

Gsn *Gsn*

Gap43 *Gap43*

Napb *Napb*

FIG. 17: Visualization of the highly-ranked local image features in discriminating genes enriched in neurons and oligodendrocytes.

An appealing property of my regularized learning method is that it can identify the SIFT features and the corresponding image patches that are highly predictive of cell-type enrichment. These highly-ranked features are expected to be located in regions where the most discriminative properties of cell-type enrichment are found, thereby distinguishing the cell-type-specificity of genes accurately. In-depth analysis of these highly-discriminative features might help elucidating the relationships among different brain cell-types. To this end, I obtain and visualize the highly-ranked features for classifying genes enriched in neurons and oligodendrocytes.

Specifically, I use stability selection to rank the bag-of-words features, which correspond to the cluster centers of the descriptor pool. Since the cluster centers might not coincide with SIFT features, I locate the SIFT features in the pool that are closest to these cluster centers. Finally, I trace back to obtain the ISH images from which these descriptors are extracted. I also record the specific locations that these SIFT features are computed and the names of genes corresponding to these ISH images. Some sample highly-ranked features are visualized in Figure 17. For each highly-ranked feature (*i.e.*, cluster center) generated by stability selection, I found the closest SIFT descriptor in the pool and then displayed the corresponding ISH image and the locations on which the SIFT descriptor was computed. The images in the left column are the ISH images along with the SIFT descriptors. The right column shows parts of the ISH images in red boxes on the corresponding image to the left. The grid is used to illustrate the 4 by 4 neighborhoods for the SIFT descriptor. The arrow denotes the direction and the length denotes the magnitude of the orientated histogram. I can observe that most of the highly-ranked features identified by my method are indeed located around the boundaries between regions such as hippocampus and isocortex. Additionally, most of these features span the boundary between the white matter and the gray matter. It has been widely known that the main function of oligodendrocytes is to provide support and to insulate the axons of neurons. Thus, oligodendrocytes mostly occupy the white matter. In contrast, neurons are mainly located in the gray matter to control information flow within the brain. Therefore, the most discriminative features that distinguish genes enriched in neurons and oligodendrocytes should span the boundary between the gray matter and the white matter. These results demonstrate that my feature ranking method can identify locations in the brain that can distinguish genes enriched in different cell-types, thereby providing insights on the relationships among brain cell-types.

I observe that the six tasks achieve different performance, and these differences might be related to the intrinsic relationship between various brain cell-types. In order to expedite cross-task comparison, I show the performance of the six tasks on the combination of coronal and sagittal images in Figure 18. I see that the relative performance differences among the six tasks are generally consistent across the three data sets with different levels of enrichment.

I see that the classification of genes enriched in astrocytes versus the negative set yields the lowest performance on all three data sets. Indeed, astrocytes are among the least-understood brain cells currently, though they account for a high proportion

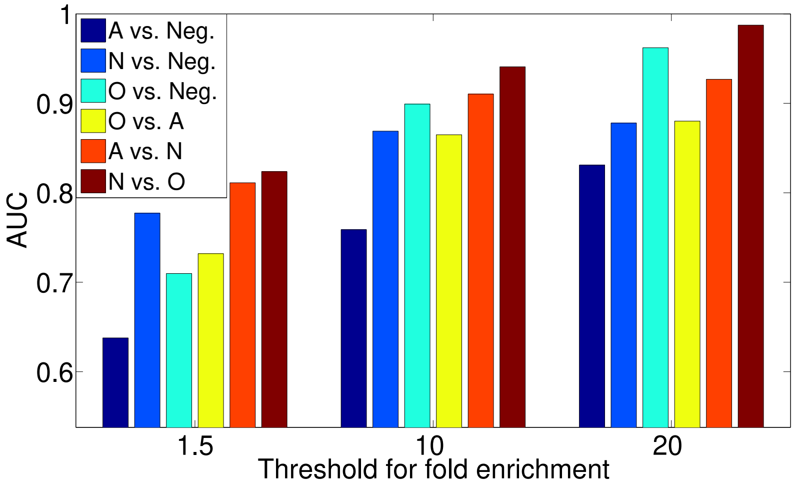


FIG. 18: Comparison of performance achieved on the six different tasks. I only report the results using combination of coronal and sagittal data, since this data yielded the best performance. The numbers of genes used for different tasks are given in Table 8.

of the brain cells [92]. This type of cells fill the space between neurons and were traditionally considered as providing supportive functions to neurons. However, recent studies showed that thy might control the concentration of extracellular molecules, thereby providing important regulatory functions [92–94]. Thus, the difficulty of distinguishing astrocytes with other cells might be due to the fact that they are spatially very close to other major brain cell-types, and they are found in all areas of the brain [92, 94, 95].

On the other hand, the classification of genes enriched in neurons and oligodendrocytes yields the highest performance on all three data sets. Indeed, oligodendrocytes are examples of well-understood glia in the brain. Their primary function is to insulate the axon and thus expedite the transduction of impulses between neurons by creating the myelin sheath [92, 94, 95]. Thus, oligodendrocytes mainly reside in the white matter, while neurons mainly reside in the gray matter. The spatial complementarity between oligodendrocytes and neurons might explain the relatively high performance of distinguishing genes enriched in these two cell-types.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this chapter, I summarize the contributions of this dissertation and discuss future research directions.

6.1 SUMMARY OF CONTRIBUTIONS

The major theme of this dissertation is to demonstrate that computational methods can be used to facilitate and speedup analysis of brain images. I propose computational approaches for completing and integrating multi-modality neuroimaging data. I then design algorithms for identifying cell-type-specific genes in the mouse brain automatically. I further provide efficient models for improving the neuron tracing performance on microscopy images. I finally develop effective methods for identifying the secondary structure elements of proteins on electronic microscopy images.

For completing and integrating multi-modality neuroimaging data, I develop a 3-D CNN model which takes one volumetric data modality as input and another modality as output. The nonlinear relationship between different data modalities is captured by a large number of trainable parameters in the network. I apply this model to predict the missing PET patterns from the MRI data. Results show that the predicted PET data achieves similar classification performance as the true PET images. Additionally, my data completion method significantly outperforms prior methods in all experiments.

In order to improve neuronal reconstruction based on microscopy images, I employ novel deep architectures to segment the neuronal voxels. The CNNs use the original 3D microscopy images as input and generate the segmentation maps as output. I compare the performance of my approach with that of commonly used tracing methods. Results show that the proposed model significantly outperforms prior methods on other neuron tracing methods on microscopy images. Overall, it is demonstrated that the proposed method produces more accurate results on neuronal morphology reconstruction.

The identification of secondary structure of proteins is challenging because of their structural similarities in 3D space. In order to extract the essential characteristics of secondary structure elements, I propose to use 3D CNNs to segment the EM images. The proposed CNNs use the 3D EM density maps as inputs and generate the classification labels as outputs. Then I employ the post processing step for improving the obtained segmentation maps. I compare the performance of my method with prior learning based methods for protein structure annotation. Results show that the performance of the proposed method is significantly better than the existing methods.

The automatical identification of cell-type-specific genes in the mouse brain is achieved by combining the high-resolution ISH images from the Allen Brain Atlas with the experimentally-generated lists of genes enriched in astrocytes, neurons, and oligodendrocytes. I construct invariant, high-level representations from the ISH images directly and employ advanced machine learning techniques to perform the classification and image feature selection. Results show that my image-based representations are predictive of cell-type enrichment. I also show that the highly-ranked image features identified by my method explain the intrinsic relationships among brain cell-types. Overall, my results demonstrate that automated image computing could lead to more quantitative and accurate computational modeling and results [96–98].

6.2 FUTURE DIRECTIONS

In the multi-modality image data completion problem, I consider the CNN model in this dissertation. There are also other deep architectures that achieved promising performance on image-related tasks. It would be interesting to extend other deep models, such as the deep belief networks, for volumetric image data completion. In this dissertation, I employ a CNN model with two hidden layers due to the high computational cost of training. I will explore ways of expediting the computation and design more complicated deep models in the future.

In the segmentation problem for improving neuron tracing, there are also other deep architectures that achieved promising performance on image-related tasks. It would be interesting to apply other deep models for 3D image segmentation. For example, recent studies showed that recurrent neuron networks (RNNs) yielded very promising performance on visual recognition tasks. I will explore RNNs in the field of neuronal reconstruction in the future. In the current experiments, I only use a small number of images for training the CNN models. Prior studies have shown that the success of deep learning models relies on a large training data set. As more and more data with manual reconstruction are collected in the BigNeuron project, I will explore CNNs with deeper architectures in the future. The current work used CNN models for neuronal image segmentation and improved the quality of subsequent reconstruction. I will explore the possibility of using CNNs on neuron tracing directly in the future,.

In the protein EM images, I use only a small number of simulated density maps with medium resolution of 8Å. I will develop novel CNN models with more efficient architectures on a large scale of EM images having multiple resolutions to demonstrate the broad application potential of deep learning methods in the protein structure detection field. In addition, the detection methods should be further tested using experimentally derived EM images whose backbone structures are known. However, due to the lack of such paired data, the proposed method is currently tested only using simulated volumetric density maps. In order to show the effectiveness of the proposed method more clearly, I will explore the method on the experimentally derived EM images in the future.

In the problems of identifying cell-type-specific genes, the features for identifying cell-type-specific genes are generic representations and are not trained and tuned to specific tasks. I will explore deep models that are trained end-to-end for fully automated cell-type-specific gene prediction [9, 99]. I formulate the cell-type-specific gene identification problem into six separate classification tasks in the current work. However, the prediction of specificity in multiple cell-types might be related. I will employ multi-task learning techniques [100–102] to identify cell-type-specific genes in multiple cell-types simultaneously in the future.

REFERENCES

[1] M. W. Weiner and *et al.*, "The Alzheimers Disease Neuroimaging Initiative: A review of papers published since its inception," *Alzheimer's & Dementia*, vol. 8, no. 1, Supplement, pp. S1–S68, 2012.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, November 1998.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, pp. 1106–1114, 2012.

[4] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 769–776, 2009.

[5] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *MICCAI*, vol. 2, pp. 411–418, 2013.

[6] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung, "Convolutional networks can learn to generate affinity graphs for image segmentation," *Neural Computation*, vol. 22, no. 2, pp. 511–538, 2010.

[7] D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[8] I. G. Y. Bengio and A. Courville, “Deep learning.” Book in preparation for MIT Press, 2016.

[9] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[10] L. Yuan, Y. Wang, P. M. Thompson, V. A. Narayan, and J. Ye, “Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data,” *NeuroImage*, vol. 61, no. 3, pp. 622–632, 2012.

[11] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson, “Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model,” *Bioinformatics*, vol. 26, no. 12, pp. i38–i46, 2010.

[12] Y. Wan, F. Long, L. Qu, H. Xiao, M. Hawrylycz, E. W. Myers, and H. Peng, “BlastNeuron for automated comparison, retrieval and clustering of 3d neuron morphologies,” *Neuroinformatics*, vol. 13, no. 4, pp. 487–499, 2015.

[13] T. Smafield, V. Pasupuleti, K. Sharma, R. L. Huganir, B. Ye, and J. Zhou, “Automatic dendritic length quantification for high throughput screening of mature neurons,” *Neuroinformatics*, vol. 13, no. 4, pp. 443–458, 2015.

[14] Z. Zhou, S. A. Sorensen, and H. Peng, “Neuron crawler: an automatic tracing algorithm for very large neuron images,” in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*, pp. 870–874, IEEE, 2015.

[15] Z. Zhou, X. Liu, B. Long, and H. Peng, “TReMAP: Automatic 3d neuron reconstruction based on tracing, reverse mapping and assembling of 2d projections,” *Neuroinformatics*, vol. 14, no. 1, pp. 41–50, 2016.

[16] L. Gu and L. Cheng, “Learning to boost filamentary structure segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, p-p. 639–647, 2015.

[17] H. Xiao and H. Peng, “APP2: automatic tracing of 3d neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree,” *Bioinformatics*, vol. 29, no. 11, pp. 1448–1454, 2013.

[18] J. Yang, P. T. Gonzalez-Bellido, and H. Peng, “A distance-field based automatic neuron tracing method,” *BMC bioinformatics*, vol. 14, no. 1, p. 1, 2013.

[19] H. Chen, H. Xiao, T. Liu, and H. Peng, “SmartTracing: self-learning-based neuron reconstruction,” *Brain Informatics*, vol. 2, no. 3, pp. 135–144, 2015.

[20] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Computer vision–ECCV 2014*, pp. 818–833, Springer, 2014.

[21] T. Zeng, R. Li, R. Mukkamala, J. Ye, and S. Ji, "Deep convolutional neural networks for annotating gene expression patterns in the mouse brain," *BMC bioinformatics*, vol. 16, no. 1, p. 1, 2015.

[22] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II–97, IEEE, 2004.

[23] W. Zhang, R. Li, H. Deng, L. Wang, W. Lin, S. Ji, and D. Shen, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214–224, 2015.

[24] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli, "BigNeuron: large-scale 3d neuron reconstruction from optical microscopy images," *Neuron*, vol. 87, no. 2, pp. 252–256, 2015.

[25] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, "V3D enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnology*, vol. 28, no. 4, pp. 348–353, 2010.

[26] A. Giusti, D. C. Cireşan, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," *arXiv preprint arXiv:1302.1700*, 2013.

[27] H. Li, R. Zhao, and X. Wang, “Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification,” *arXiv preprint arXiv:1412.4526*, 2014.

[28] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

[29] H. Chen, X. Qi, J.-Z. Cheng, and P.-A. Heng, “Deep contextual networks for neuronal structure segmentation,” in *Thirtyth AAAI Conference on Artificial Intelligence*, pp. 000–000, 2015.

[30] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” *arXiv preprint arXiv:1505.07293*, 2015.

[31] R. Li, W. Zhang, H.-I. Suk, L. Wang, J. Li, D. Shen, and S. Ji, “Deep learning based imaging data completion for improved brain disease diagnosis,” in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014*, pp. 305–312, Springer, 2014.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[33] C. Szegedy, S. Ioffe, and V. Vanhoucke, “Inception-v4, inception-resnet and the impact of residual connections on learning,” *arXiv preprint arXiv:1602.07261*, 2016.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

[35] W. Jiang, M. L. Baker, S. J. Ludtke, and W. Chiu, “Bridging the information gap: computational tools for intermediate resolution structure interpretation,” *Journal of molecular biology*, vol. 308, no. 5, pp. 1033–1044, 2001.

[36] M. L. Baker, T. Ju, and W. Chiu, “Identification of secondary structure elements in intermediate-resolution density maps,” *Structure*, vol. 15, no. 1, pp. 7–19, 2007.

[37] Z. Yu and C. Bajaj, “Computational approaches for automatic structural analysis of large biomolecular complexes,” *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. 5, no. 4, pp. 568–582, 2008.

[38] Y. Kong and J. Ma, “A structural-informatics approach for mining β-sheets: locating sheets in intermediate-resolution density maps,” *Journal of molecular biology*, vol. 332, no. 2, pp. 399–413, 2003.

[39] S. J. Ludtke, M. L. Baker, D.-H. Chen, J.-L. Song, D. T. Chuang, and W. Chiu, “De novo backbone trace of groel from single particle electron cryomicroscopy,” *Structure*, vol. 16, no. 3, pp. 441–448, 2008.

[40] L. Ma, M. Reisert, and H. Burkhardt, “Rennsh: a novel alpha-helix identification approach for intermediate resolution electron density maps,” *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. 9, no. 1, pp. 228–239, 2012.

[41] D. Si, S. Ji, K. A. Nasr, and J. He, “A machine learning approach for the identification of protein secondary structure elements from electron cryo-microscopy density maps,” *Biopolymers*, vol. 97, no. 9, pp. 698–708, 2012.

[42] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[43] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” *arXiv preprint arXiv:1412.7062*, 2014.

[44] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.

[45] A. Fakhry, T. Zeng, and S. Ji, “Residual deconvolutional networks for neurite segmentation from electron microscopy images,” *IEEE Transactions on Medical Imaging*, submitted.

[46] C. C. Huang, G. S. Couch, E. F. Pettersen, and T. E. Ferrin, “Chimera: an extensible molecular modeling application constructed using standard components,” in *Pacific symposium on biocomputing*, vol. 1, pp. 1519–1523, 1996.

[47] J. D. Cahoy, B. Emery, A. Kaushal, L. C. Foo, J. L. Zamanian, K. S. Christopherson, Y. Xing, J. L. Lubischer, P. A. Krieg, S. A. Krupenko, W. J. Thompson, and B. A. Barres, "A transcriptome database for astrocytes, neurons, and oligodendrocytes: A new resource for understanding brain development and function," *The Journal of Neuroscience*, vol. 28, no. 1, pp. 264–278, 2008.

[48] P. Grange, M. Hawrylycz, and P. P. Mitra, "Cell-type-specific microarray data and the Allen atlas: quantitative analysis of brain-wide patterns of correlation and density," *arXiv:1303.0013 [q-bio.NC]*, 2013.

[49] B. W. Okaty, K. Sugino, and S. B. Nelson, "A quantitative comparison of cell-type-specific microarray gene expression profiling methods in the mouse brain," *PLoS One*, vol. 6, no. 1, p. e16493, 2011.

[50] Y. Ko, S. A. Ament, J. A. Eddy, J. Caballero, J. C. Earls, L. Hood, and N. D. Price, "Cell type-specific genes show striking and distinct patterns of spatial expression in the mouse brain," *Proceedings of the National Academy of Sciences*, vol. 110, no. 8, pp. 3095–3100, 2013.

[51] L. French, P. P. C. Tan, and P. Pavlidis, "Large-scale analysis of gene expression and connectivity in the rodent brain: insights through data integration," *Frontiers in Neuroinformatics*, vol. 5, no. 12, 2011.

[52] P. P. C. Tan, L. French, and P. Pavlidis, "Neuron-enriched gene expression patterns are regionally anti-correlated with oligodendrocyte-enriched patterns

in the adult mouse and human brain," *Frontiers in Neuroscience*, vol. 7, no. 5, 2013.

[53] P. Grange and P. Mitra, "Computational neuroanatomy and gene expression: Optimal sets of marker genes for brain regions," in *Proceedings of the 46th Annual Conference on Information Sciences and Systems*, (Princeton,NJ,USA), pp. 1–6, IEEE, 2012.

[54] S. Ji, "Computational genetic neuroanatomy of the developing mouse brain: dimensionality reduction, visualization, and clustering," *BMC Bioinformatics*, vol. 14, p. 222, 2013.

[55] S. Ji, A. Fakhry, and H. Deng, "Integrative analysis of the connectivity and gene expression atlases in the mouse brain," *NeuroImage*, vol. 84, no. 1, pp. 245–253, 2014.

[56] N. S. Zuckerman, Y. Noam, A. J. Goldsmith, and P. P. Lee, "A self-directed method for cell-type identification and separation of gene expression microarrays," *PLoS computational biology*, vol. 9, no. 8, p. e1003189, 2013.

[57] M. C. Oldham, G. Konopka, K. Iwamoto, P. Langfelder, T. Kato, S. Horvath, and D. H. Geschwind, "Functional organization of the transcriptome in human brain," *Nature Neuroscience*, vol. 11, no. 11, pp. 1271–1282, 2008.

[58] M. J. Hawrylycz and *et al.*, "An anatomically comprehensive atlas of the adult human brain transcriptome," *Nature*, vol. 489, no. 7416, pp. 391–399, 2012.

[59] E. S. Lein and *et al.*, “Genome-wide atlas of gene expression in the adult mouse brain,” *Nature*, vol. 445, no. 7124, pp. 168–176, 2007.

[60] S. Ganguli and H. Sompolinsky, “Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis,” *Annual Review of Neuroscience*, vol. 35, no. 1, pp. 485–508, 2012.

[61] D. M. Green and J. A. Swets, *Signal detection theory and psychophysics*. New York, NY, USA: John Wiley and Sons Inc., 1 ed., 1966.

[62] K. A. Spackman, “Signal detection theory: valuable tools for evaluating inductive learning,” in *Proceedings of the sixth international workshop on Machine learning*, (San Francisco, CA, USA), pp. 160–163, Morgan Kaufmann Publishers Inc., 1989.

[63] Allen Institute for Brain Science, “Allen Mouse Brain Atlas [Internet],” 2012.

[64] H.-W. Dong, *The Allen Reference Atlas: A Digital Color Brain Atlas of the C57BL/6J Male Mouse*. Hoboken,NJ: John Wiley & Sons Inc., 2009.

[65] Allen Institute for Brain Science, “Allen Mouse Brain Atlas: Technical White Paper: Informatics Data Processing,” 2012.

[66] L. Ng, S. Pathak, C. Kuan, C. Lau, H. Dong, A. Sodt, C. Dang, B. Avants, P. Yushkevich, J. Gee, D. Haynor, E. Lein, A. Jones, and M. Hawrylycz, “Neuroinformatics for genome-wide 3-D gene expression mapping in the mouse

brain," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, pp. 382–393, 2007.

[67] L. Ng and *et al*, "An anatomic gene expression atlas of the adult mouse brain," *Nature Neuroscience*, vol. 12, no. 3, pp. 356–362, 2009.

[68] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[69] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[70] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proceedings of the 9th European Conference on Computer Vision*, (Berlin, Heidelberg), pp. 490–503, Springer-Verlag, 2006.

[71] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Proceedings of the IEEE 11th International Conference on Computer Vision*, (Rio de Janeiro, Brazil), pp. 1–8, IEEE Computer Society, 2007.

[72] N. Liscovitch, U. Shalit, and G. Chechik, "FuncISH: learning a functional representation of neural ISH images," *Bioinformatics*, vol. 29, no. 13, pp. i36–i43, 2013.

[73] S. Ji, L. Sun, R. Jin, S. Kumar, and J. Ye, "Automated annotation of *Drosophila* gene expression patterns using a controlled vocabulary," *Bioinformatics*, vol. 24, no. 17, pp. 1881–1888, 2008.

[74] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms." `http://www.vlfeat.org/`, 2008.

[75] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *ECCV Workshop on Statistical Learning in Computer Vision*, vol. 1, (Prague, Czech Republic,), pp. 1–22, 2004.

[76] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Diego,CA, USA), pp. 524–531, IEEE Computer Society,, 2005.

[77] S. Lazebnik, C. Achmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (New York, NY, USA), pp. 2169–2178, IEEE Computer Society,, 2006.

[78] S. Ji, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye, "A bag-of-words approach for *Drosophila* gene expression pattern annotation," *BMC Bioinformatics*, vol. 10, no. 1, p. 119, 2009.

[79] L. Kirsch, N. Liscovitch, and G. Chechik, “Localizing genes to cerebellar layers by classifying ISH images,” *PLOS computational biology*, vol. 8, no. 12, p. e1002790, 2012.

[80] S. Ryali, K. Supekar, D. A. Abrams, and V. Menon, “Sparse logistic regression for whole-brain classification of fMRI data,” *NeuroImage*, vol. 51, no. 2, pp. 752 – 764, 2010.

[81] M. de Brecht and N. Yamagishi, “Combining sparseness and smoothness improves classification accuracy and interpretability,” *NeuroImage*, vol. 60, no. 2, pp. 1550 – 1561, 2012.

[82] C.-J. Lin, R. C. Weng, and S. S. Keerthi, “Trust region newton method for logistic regression,” *Journal of Machine Learning Research*, vol. 9, pp. 627–650, 2008.

[83] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[84] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, “Recent advances of large-scale linear classification,” *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2584–2603, 2012.

[85] J. Liu, S. Ji, and J. Ye, “SLEP: Sparse Learning with Efficient Projections,” *Arizona State University*, 2009. http://www.public.asu.edu/˜jye02/Software/SLEP/.

[86] G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "A comparison of optimization methods and software for large-scale L1-regularized linear classification," *Journal of Machine Learning Research*, vol. 11, pp. 3183–3234, 2010.

[87] F. Xing, H. Su, J. Neltner, and L. Yang, "Automatic ki-67 counting using robust cell detection and online dictionary learning," *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 859–870, March 2014.

[88] H. Su, F. Xing, J. Lee, C. Peterson, and L. Yang, "Automatic myonuclear detection in isolated single muscle fibers using robust ellipse fitting and sparse optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. PP, no. 99, pp. 1–1, 2013.

[89] P. Bühlmann, "Bagging, boosting and ensemble methods," in *Handbook of Computational Statistics: Concepts and Methods* (J. Gentle, W. Härdle, and Y. Mori, eds.), Springer Handbooks of Computational Statistics, pp. 877–907, Berlin: Springer, 2004.

[90] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[91] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[92] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of Neural Science*. New York: McGraw-Hill Professional, fifth ed., 2012.

[93] W. Walz, "Role of astrocytes in the clearance of excess extracellular potassium," *Neurochemistry International*, vol. 36, no. 4-5, pp. 291–300, 2000.

[94] M. F. Bear, B. W. Connors, and M. A. Paradiso, *Neuroscience: Exploring the Brain*. Baltimore, MD,USA: Lippincott Williams & Wilkins, third ed., 2006.

[95] C. Watson, M. Kirkcaldie, and G. Paxinos, *The Brain: An Introduction to Functional Neuroanatomy*. Academic Press, San Diego, CA, USA, 1 ed., 2010.

[96] H. Peng, B. Roysam, and G. Ascoli, "Automated image computing reshapes computational neuroscience," *BMC Bioinformatics*, vol. 14, no. 1, p. 293, 2013.

[97] H. Peng, "Bioimage informatics: a new area of engineering biology," *Bioinformatics*, vol. 24, no. 17, pp. 1827–1836, 2008.

[98] R. Ugolotti, P. Mesejo, S. Zongaro, B. Bardoni, G. Berto, F. Bianchi, I. Molineris, M. Giacobini, S. Cagnoni, and F. Di Cunto, "Visual search of neuropil-enriched RNAs from brain in situ hybridization data through the image analysis pipeline Hippo-ATESC," *PlOS ONE*, vol. 8, no. 9, p. e74481, 2013.

[99] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[100] T. K. Pong, P. Tseng, S. Ji, and J. Ye, “Trace norm regularization: Reformulations, algorithms, and multi-task learning,” *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3465–3489, 2010.

[101] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, (Montreal, Canada), Association for Uncertainty in Artificial Intelligence.

[102] D. Zhang and D. Shen, “Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in Alzheimer’s disease,” *NeuroImage*, vol. 59, no. 2, pp. 895–907, 2012.

VITA

Rongjian Li

Department of Computer Science

Old Dominion University

Norfolk, VA 23529

I received the Bachelor and Master degrees in Applied Mathematics from Xi'an Jiaotong University, Xi'an, China, in 2007 and 2012, respectively. In Fall 2012, I joined in Computer Science Department of Old Dominion University and started my research in machine learning, data mining and computational biology. I have worked on machine learning methods for representing and analysing gene expression pattern images in fruit fly embryo and mouse brain. I have also worked on deep learning method and its application for 2D and 3D medical and biological images. I have already finished four papers in highly-regarded conference and journal, and I serve as the primary or co-author of a total of eleven papers.

Typeset using LaTeX.