# DEEP LEARNING APPLICATIONS IN MAGNETIC RESONANCE IMAGING

by

BRANDON JAMES CAMPBELL

(Under the Direction of Qun Zhao)

## ABSTRACT

Magnetic resonance imaging (MRI) is a non-invasive and versatile imagining technique with applications ranging from tumor detection to brain functional imaging. Recent successes with Deep Learning algorithms, specifically convolutional neural networks (CNNs), in image classification and segmentation tasks have led to significant research in determining the capabilities of extending these algorithms to MRI. In this thesis, we explore CNN capabilities with two distinctly different tasks: classification of adipose tissue and brain functional connectivity analysis.

For adipose tissue classification, we find that not only are CNNs capable of achieving state of the art accuracies based on pre-processed datasets, but these accuracies hold when used upon the raw complex signals derived from the MRI scanner. For brain functional connectivity analysis, we explore a new method for discovering resting state networks in pig brains. Pigs are a popular research replacement for human subjects, but brain functional connectivity is largely unexplored in these animals.

INDEX WORDS: magnetic resonance imaging, machine learning, deep learning, convolutional neural networks, water-fat separation, image classification, brain functional analysis, functional magnetic resonance imaging, bold imaging

# DEEP LEARNING APPLICATIONS IN MAGNETIC RESONANCE IMAGING

by

BRANDON JAMES CAMPBELL

BS, Clemson University, 2014

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2017

DEEP LEARNING APPLICATIONS IN MAGNETIC RESONANCE IMAGING

by

BRANDON JAMES CAMPBELL

| | |
|---|---|
| Major Professor: | Qun Zhao |
| Committee: | Loris Magnani |
| | Inseok Song |

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2017

# DEDICATION

I would like to dedicate this thesis to my parents, Stanley and Gerrie Campbell. Without their continual love and support, not only would this work not exist, but I would have never had the courage to achieve my goals.

# ACKNOWLEDGEMENTS

I would like to thank my advisor, Qun Zhao, for his guidance and inspiring me to enter the field of machine learning. Instead of simply work, this field has turned into a passion that I will be able to enjoy for the rest of my life. I would also like to thank my committee, Loris Magnani and Inseok Song for their guidance and patience throughout my academic time at UGA.

Greg Simchick and Alice Shen also deserve a multitude of thanks. Without Greg's expertise at running the MRI scanner, I would not have had much of the data presented in this work. He has also been a great friend during the long hours at work, and his conversation will be sorely missed. Through discussion, Alice was able to help me more fully understand my work. I also appreciate her help with the arduous task of analyzing gigabytes of data. Kim Mason and Shan-Ho Tsai also deserve thanks for their help in the BIRC and the GACRC respectively. Dr. Franklin West and his group also deserve thanks for data acquisition and support.

Finally, I thank my family and friends. So many things in life are individual responsibilities but their companionship, encouragement and love truly make these things possible and worth doing. I especially want to thank my loving, supportive girlfriend Lauren Ashley Eserman. She has given me so many reasons to continue when I would rather give up, and I will forever be grateful.

# TABLE OF CONTENTS

Page

# LIST OF TABLES

# LIST OF FIGURES

Page

# CHAPTER 1

# INTRODUCTION

## Motivation

Magnetic resonance imaging (MRI) is one of the most important imaging techniques in both clinical and research applications due to its non-invasive nature and versatility. These two aspects have allowed MRI a myriad of different applications ranging from tumor detection to brain functional imaging [1]–[4]. However, each of these applications require, at least in some part, the use of selective contrast and post-processing of MRI signals to classify or segment the interesting aspects in MRI images, and discovering the best methods to create these contrasts is currently a very popular research topic [5], [6]. There is an entire medical discipline known as Radiology that is devoted in part to understanding what can be visualized in MRI images. In recent years the success of machine learning algorithms, specifically convolutional neural networks, in classification, segmentation and functional connectivity of images has led to significant research in determining the capabilities of extending these algorithms to MRI [7]–[11]. In this thesis two such projects are presented that explore these capabilities in MRI for two distinctly different tasks:  classification of adipose tissues and brain functional connectivity analysis.

## MRI Overview

MRI signals are derived from the interaction of atomic nuclear spins with an external magnetic field, $\overrightarrow{B_0}$ [12]. Most often the atom of interest is hydrogen as it is the most abundant atomic species present in biological tissues. In the presence of an external magnetic field, slightly more than 50% of protons will precess about an axis approximately[1] parallel to the external magnetic field vector $\overrightarrow{B_0}$, while slightly less than 50% of protons precess anti-parallel to $\overrightarrow{B_0}$. The angular frequency of this precession is given by the Larmor frequency equation,

$$\omega = \gamma B_0 \qquad (1)$$

where $\omega$ is the angular frequency of the proton spin in radians per second, and $\gamma$ is the gyromagnetic ratio. The summation of magnetic moments for every proton in a volume provides a net magnetic moment or magnetization, $\overrightarrow{M}$, which again is approximately parallel to the external magnetic field vector [12]. If $\overrightarrow{B_0}$ is applied in the positive z-direction, then this can be expressed as $\vec{M} = M_0\hat{z}$, where $M_0$ is initial magnetization.

In the presence of external magnetic fields common in MRI (between 1.5T – 9T), the Larmor frequency of a hydrogen proton is within radio-frequency (RF) range. By providing a short RF-pulse that generates a magnetic field $\overrightarrow{B_1}$, which is perpendicular to $\overrightarrow{B_0}$ and oscillates at the Larmor frequency of hydrogen, the magnetization of a volume will begin to precess simultaneously about the two axes, $\overrightarrow{B_0}$ and $\overrightarrow{B_1}$ [12]. Each of the proton magnetic moments will precess in a plane transverse to the original magnetization. Once the RF-pulse has been removed this transverse magnetization vector will begin to return to the original z-direction or longitudinal direction. This return of $\vec{M}$ is described by the Bloch equation,

$$\frac{d\vec{M}}{dt} = \gamma\vec{M} \times \overrightarrow{B_0} + \frac{1}{T_1}(M_0 - M_z)\hat{z} - \frac{1}{T_2}\overrightarrow{M_{xy}} \quad (2)$$

[1] The specific angle between the magnetic moment vector and external magnetic field is affected by the local chemical environment of an atom.

where $T_1$is the longitudinal relaxation time, $M_z$ is the z-component of the magnetization, $T_2$ is the transverse relaxation time, and $M_{xy}$ is the transverse magnetization [12]. Solving the Bloch equation for the longitudinal and transverse directions provides equations for the behavior of the magnetization in these directions directly after the RF pulse. These equations take the form:

$$M_z(t) = M_0(1 - e^{-\frac{t}{T_1}}) \qquad (3)$$

$$M_{xy}(t) = M_{xy}(0)e^{-\frac{t}{T_2}} \qquad (4)$$

From **Equation (3)**, directly after the RF-pulse (time zero) the magnetization in the z-direction is zero and as time increases the magnetization increases. **Equation (4)** shows the opposite trend, such that at time zero the magnetization in the transverse plane is maximized and decreases as time increases. Due to the nonzero component of the magnetization oscillating in the transverse plane, an electromotive force can be detected as an oscillating voltage signal by coils surrounding the volume. This oscillating voltage decreases exponentially as the magnetization relaxes longitudinally, which can be seen in **Equation (4)**, and is known as free induction decay (FID) [12]. From this FID signal, proton density, $T_1$, and $T_2$ can be determined. Additionally, the spatial location of these values can be obtained through a process of Fourier encoding. The decomposition of the FID signal into different important signal features such as $T_2$, and chemical spectroscopic peak parameters is the basis for much of MRI signal analysis.

## Machine Learning, Convolutional Neural Networks, and Deep Learning Overview

Machine learning is a field of computer science that attempts to create algorithms that can learn to solve a specific task without being specifically programed to solve that task. In a broad sense, these types of algorithms tend to be iterative statistical methods where the success of an algorithm is based upon how accurately the algorithm can make predictions upon sets of data.

While many different examples of machine learning models have been constructed, the fully connected neural network and convolutional neural network, which will be explained in more detail, have become some of the most popular models due to their wide range of applicability and accuracy. Additionally, variations of these models have been created for specialized tasks. Deep Learning focuses on chaining multiple layers of these models in succession.

Convolutional neural networks (CNNs) are a type of deep-learning, feed-forward artificial neural network model that have been shown to be highly successful in image classification and segmentation [13], [14]. While there are multiple types of learning tasks that are possible with neural networks, this section will focus on supervised learning as a classification tool to describe the general theory behind these models. Supervised learning utilizes a set of labeled data samples that can be used to help train a model which can then be used to make predictions on unknown data samples. As with all neural networks the goal of CNNs is to solve a task by minimizing a loss function based on the transformation of a set of input data samples to an output via weights [15]. To train the model, inputs from multiple labeled samples of a training dataset are sent through the algorithm to receive an output which is then compared to the sample's label via the loss function. A simple example is the mean squared error (MSE) loss function with the form,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\widehat{Y_\iota} - Y_i)^2 \qquad (5)$$

where $\widehat{Y_\iota}$ is the samples known label, $Y_i$ is the output from the model, and n is the total number of data samples sent though the model.

The next step in the training process is to adjust each of the weights in the model to achieve a lower loss value. Most often this process is backpropagated via stochastic gradient descent (SGD) [15]. SGD is a gradient descent optimization method that utilizes a stochastic approximation for minimizing the loss function [16]. Deep learning models tend to become more accurate as the amount of input samples is increased, but using normal gradient descent methods would require updating the model weights after an entire dataset has been passed through the model. Since this would often require an unreasonable amount of computational storage, SGD instead randomly separates an entire training set into groups called *batches*, which contain the same number of samples in each batch. Once a batch is sent through the model, the loss function is calculated for only this batch, and thus an estimation of the overall loss is determined. The gradient descent method is then applied to the loss function and then each of the weights in the model are adjusted by a predetermined amount, or *learning rate*, in the direction prescribed by gradient descent. While any individual gradient calculation on a batch may not necessarily provide the ideal direction to reach a global minimum of the loss function, given that each of the samples has been selected sufficiently randomly, the general trend will be toward a minimization of the loss function after each of the batches have been completed. The term *epoch* is used to describe the state after the SGD process has iterated through each of the batches.

Often datasets are separated into a training set and a testing (or validation) set to determine the accuracy of the trained model [15]. The training set is sent through the model and used to determine the overall loss value of a dataset for the SGD process. The testing samples are not exposed to the SGD process and therefore used to determine how the model will react to unknown data. In this way the ability of the model to accurately make predictions on new datasets can be determined. For instance, if the loss value of the training set is very high but the loss value for the testing dataset is low, then the model is over trained to the training dataset and is unable to make predictions on unknown data [15]. An undertrained case would present itself when the loss values for the trained and testing datasets are both poor. In general, a deep learning model has been successfully trained when the loss value for a given set of data is sufficiently low for the task. The specific number of epochs that training may require is dataset dependent, and many different parameters, such as the learning rate, may need to be adjusted to achieve the best possible loss values.

## CNNs and Deep-Learning Architecture

The term *deep-learning* refers to the utilization of a chain of definably different layers comprised of collections of weights to complete a learning task [15]. The depth of any given model is the quantity of layers in the model.  The layers between the input and outputs in a model are referred to as hidden layers, as they require the knowledge of their connected layers to have any inherent meaning, and cannot therefore be an input or output on their own. CNNs differ from other deep-learning, artificial neural network models due to the way that layers are generated and organized to create an overall model architecture. The most common layers in CNNs are convolution layers, pooling layers, and fully connected layers.

### *Convolution Layer*

Convolution layers are based off the mathematical convolution operation [14], [15]. A sliding sequence of weights called a kernel, or filter, is multiplied via a dot product operation with data points from an input sample to produce a single valued output. **Figure 1.1A** shows a schematic representation of this process. The number of data points selected from an input sample is based off the kernel's length. In **Figure 1.1A** the kernel contains four weight values, therefore the first output of the convolution operation is the result of the first four values in the input sequence dotted with the kernel. The kernel then slides over four places, where it is then dotted with the next four input data points. The convolution process is completed once the kernel has visited every location present in the input data. The combined sequence of each convolution result is often referred to as an *activation map*.

In general, a layer of convolution contains many different kernels and therefore each input produces a number of activation maps equal to the number of kernels for each layer which can be seen in **Figure 1.1B**. The purpose behind convolution and multiple kernels is the idea that through the training process each kernel has the potential to learn an important feature present in the input data. In the case of time series data, each kernel may represent a different sub-shape within the original time series. Different time series samples can then be classified by how strongly they correlate with specific, or groups of specific, kernels.

Often, an activation function will be applied to the outputs of a layer of convolution. A common and important example is the rectified linear unit, or ReLU [17]. The ReLU is essentially a max function of the form $f(x) = \max(0, x)$. This activation function assures that the outputs of a convolution layer are positive, speeding training and avoiding the vanishing gradient problem during SGD.

### *Pooling Layer*

Pooling layers perform a down sampling of the convolution layer outputs [15], [18]. There are many types of pooling, but the two most common are *max pooling* and *average pooling*. A window of predefined size slides across the outputs (activation maps) of the convolutional layer, and the values matching the type of pooling are extracted to produce a new output. **Figure 1.2A** is a schematic representation of max pooling with a window size of 2. Pooling allows the most valuable information to be retained while reducing dimensionality, or the number of weights, and improving generalizability and computational time. **Figure 1.2B** shows an example of the results of a max pooling operation with a windows size of 2. By comparing the original input and the result of pooling it can be seen that the overall shape of the original input has been conserved.

### *Fully Connected Layer*

Fully connected layers represent the traditional notion of artificial neural networks as a collection of artificial neurons that are all connected to each other [15], [19]. **Figure 1.3A** shows a schematic of the inputs and outputs produced by a single artificial neuron. Each input is connected to the neuron via a weight with the output of the neuron being the sum of each of these values. **Figure 1.3B** demonstrates the full connection of the input dataset to the output. Each of the lines extending from a neuron are connected to each neuron in front, yet none of the neurons in a single layer are connected. This type of connection denotes a feed-forward neural network.

Ignoring the complexities of how each of the connections between layers are determined, the main difference between the convolutional layer and the fully-connected layer is that the kernels in a convolutional layer allow the model to learn spatially connected information.

## Thesis Overview

In the following chapters two applications of CNNs will be explored for analyzing MRI results. Chapter 2 will explore image segmentation with the intention of differentiating two types of adipose tissue in mice models: brown adipose tissue and white adipose tissue. This CNN implementation is also compared with other common machine learning algorithms as a means to determine the accuracy and precision of the CNN. Chapter 3 describes a new method of brain functional analysis on the pig brain. Little is currently known about the connectivity of the pig brain. Traditional methods for determining functionality in human brains requires *a priori* information to make predictions about connectivity in the brain. This type of information is lacking in pigs and therefore there is need for a new method to accomplish this task. A specific type of CNN implementation, called a sparse autoencoder, allows for the unsupervised analysis of data and the extraction of important features about datasets. This autoencoder is applied to fMRI time series data received from a pig brain and a new method of brain functional analysis is explored.

## References

[1] B. Biswal, F. Zerrin Yetkin, V. M. Haughton, and J. S. Hyde, "Functional connectivity in the motor cortex of resting human brain using echo???planar mri," *Magn. Reson. Med.*, vol. 34, no. 4, pp. 537–541, 1995.

[2] S. Ogawa, R. S. Menon, D. W. Tank, S. G. Kim, H. Merkle, J. M. Ellermann, K. Ugurbil, K. Ugurbilt, and K. Ugurbil, "Functional brain mapping by blood oxygenation level-dependent contrast magnetic resonance imaging. A comparison of signal characteristics with a biophysical model.," *Biophys. J.*, vol. 64, no. 3, pp. 803–812, 1993.

[3] X. Cao, G. Jia, T. Zhang, M. Yang, B. Wang, P. A. Wassenaar, H. Cheng, M. V. Knopp, and D. Sun, "Non-invasive MRI tumor imaging and synergistic anticancer effect of HSP90 inhibitor and glycolysis inhibitor in RIP1-Tag2 transgenic pancreatic tumor model," *Cancer Chemother. Pharmacol.*, vol. 62, no. 6, pp. 985–994, 2008.

[4] Q. Zhao, S. Lee, M. Kent, S. Schatzberg, and S. Platt, "Dynamic contrast-enhanced

magnetic resonance imaging of canine brain tumors," *Vet. Radiol. Ultrasound*, vol. 51, no. 2, pp. 122–129, 2010.

[5] W. Tang, Z. Zhen, C. Yang, L. Wang, T. Cowger, H. Chen, T. Todd, K. Hekmatyar, Q. Zhao, Y. Hou, and J. Xie, "Fe5C2 nanoparticles with high mri contrast enhancement for tumor imaging," *Small*, vol. 10, no. 7, pp. 1245–1249, 2014.

[6] Q. Zhao, J. Langley, S. Lee, and W. Liu, "Positive contrast technique for the detection and quantification of superparamagnetic iron oxide nanoparticles in MRI," *NMR Biomed.*, vol. 24, no. 5, pp. 464–472, 2011.

[7] 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.

[8] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. De Vries, M. J. N. L. Benders, and I. Isgum, "Automatic Segmentation of MR Brain Images with a Convolutional Neural Network," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1252–1261, 2016.

[9] J. Kleesiek, G. Urban, A. Hubert, D. Schwarz, K. Maier-Hein, M. Bendszus, and A. Biller, "Deep MRI brain extraction: A 3D convolutional neural network for skull stripping," *Neuroimage*, vol. 129, pp. 460–469, 2016.

[10] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 8150 LNCS, no. PART 2, pp. 246–253.

[11] A. Pinto, V. Alves, and C. A. Silva, "Brain Tumor Segmentation using Convolutional Neural Networks in MRI Images," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1240–1251,

2016.

[12] E. M. Haacke, R. W. Brown, M. R. Thompson, and R. Venkatesan, *Magnetic Resonance Imaging: Physical Principles and Sequence Design*, vol. 1st. 1999.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[15] A. Goodfellow, Ian, Bengio, Yoshua, Courville, “Deep Learning,” *MIT Press*, 2016. [Online]. Available: http://www.deeplearningbook.org/.

[16] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” *Proc. COMPSTAT’2010*, pp. 177–186, 2010.

[17] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” *AISTATS ’11 Proc. 14th Int. Conf. Artif. Intell. Stat.*, vol. 15, pp. 315–323, 2011.

[18] D. Scherer, A. Müller, and S. Behnke, “Evaluation of pooling operations in convolutional architectures for object recognition,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, vol. 6354 LNCS, no. PART 3, pp. 92–101.

[19] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain.,” *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958.

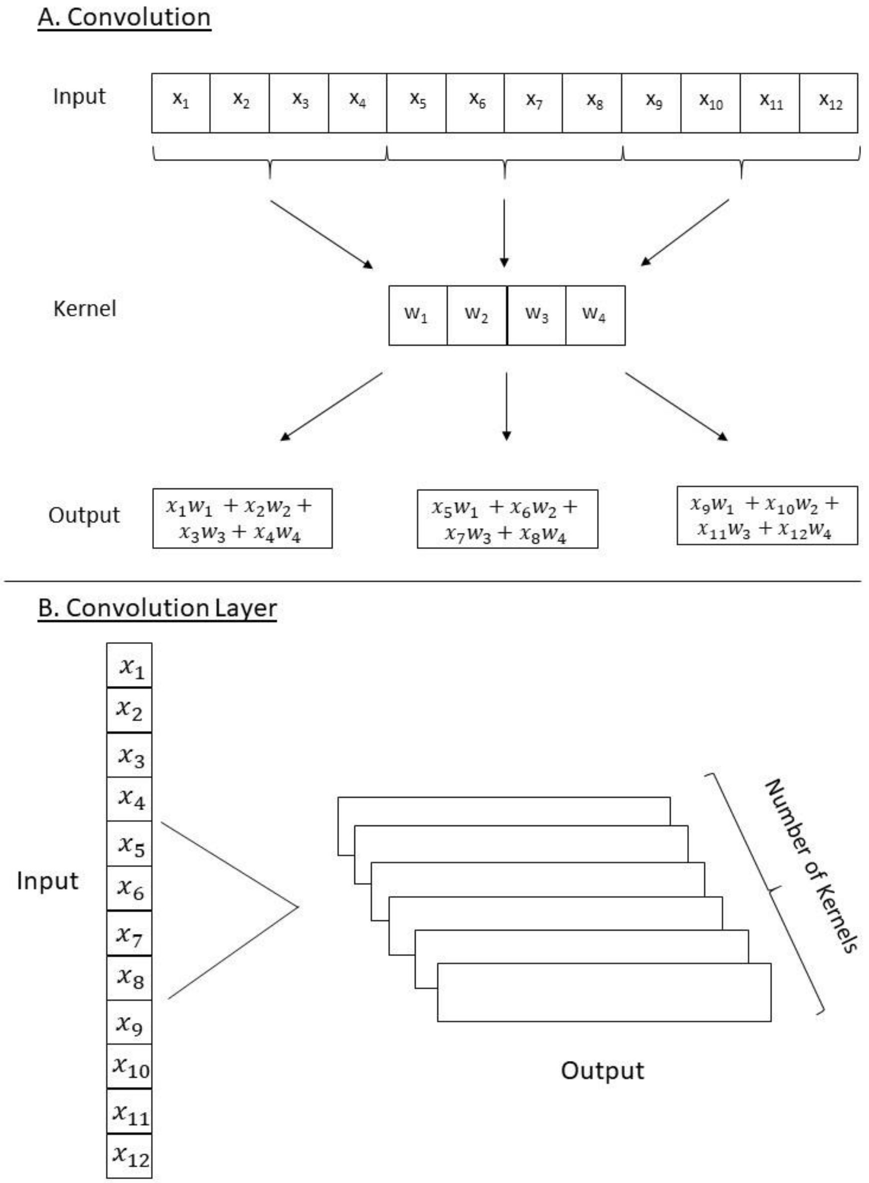
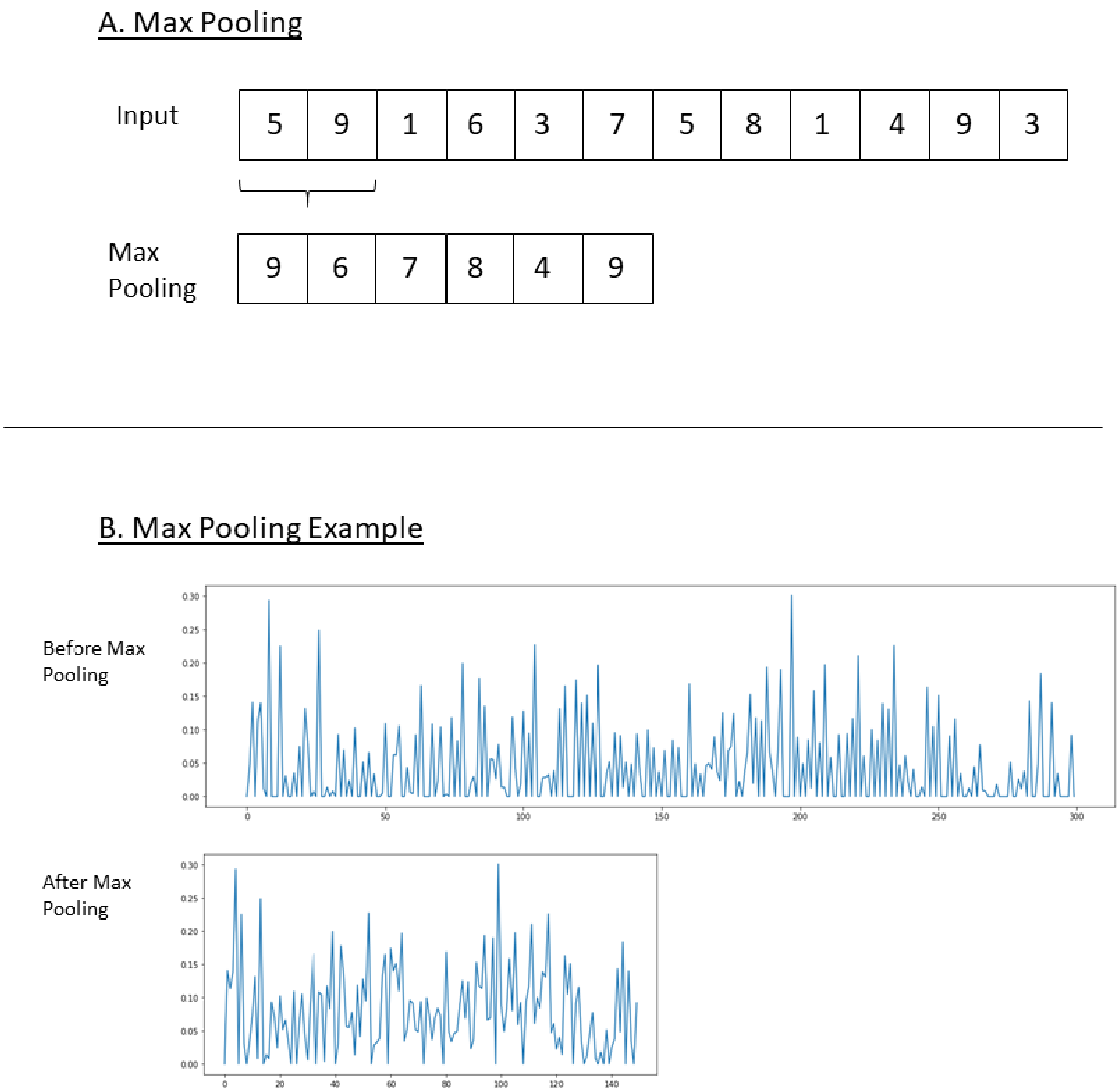

A. Convolution
Input
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$ $x_{11}$ $x_{12}$
Kernel
$w_1$ $w_2$ $w_3$ $w_4$
Output
$x_1w_1 + x_2w_2 + x_3w_3 + x_4w_4$
$x_5w_1 + x_6w_2 + x_7w_3 + x_8w_4$
$x_9w_1 + x_{10}w_2 + x_{11}w_3 + x_{12}w_4$
B. Convolution Layer
Input
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ $x_9$ $x_{10}$ $x_{11}$ $x_{12}$
Number of Kernels
Output

Figure 1.1: A.) Schematic representation of the convolution process. A sliding window, or kernel, of weight values is applied to input data via the dot product to produce outputs for each position. In this schematic a sliding window with a stride of 4 is applied to produce three outputs. B.) Schematic representation of multiple kernels. Each kernel provides a unique activation map as output.


A. Max Pooling

B. Max Pooling Example

Figure 1.2: A.) Schematic representation of the max pooling process. A sliding window of length 2 strides over the entire length of the input. Only the maximum value is retained for each stride, effectively reducing the dimension of the original input by half. B.) Example series before and after max pooling. The series after max pooling retains a similar shape even though half of the values have been removed.

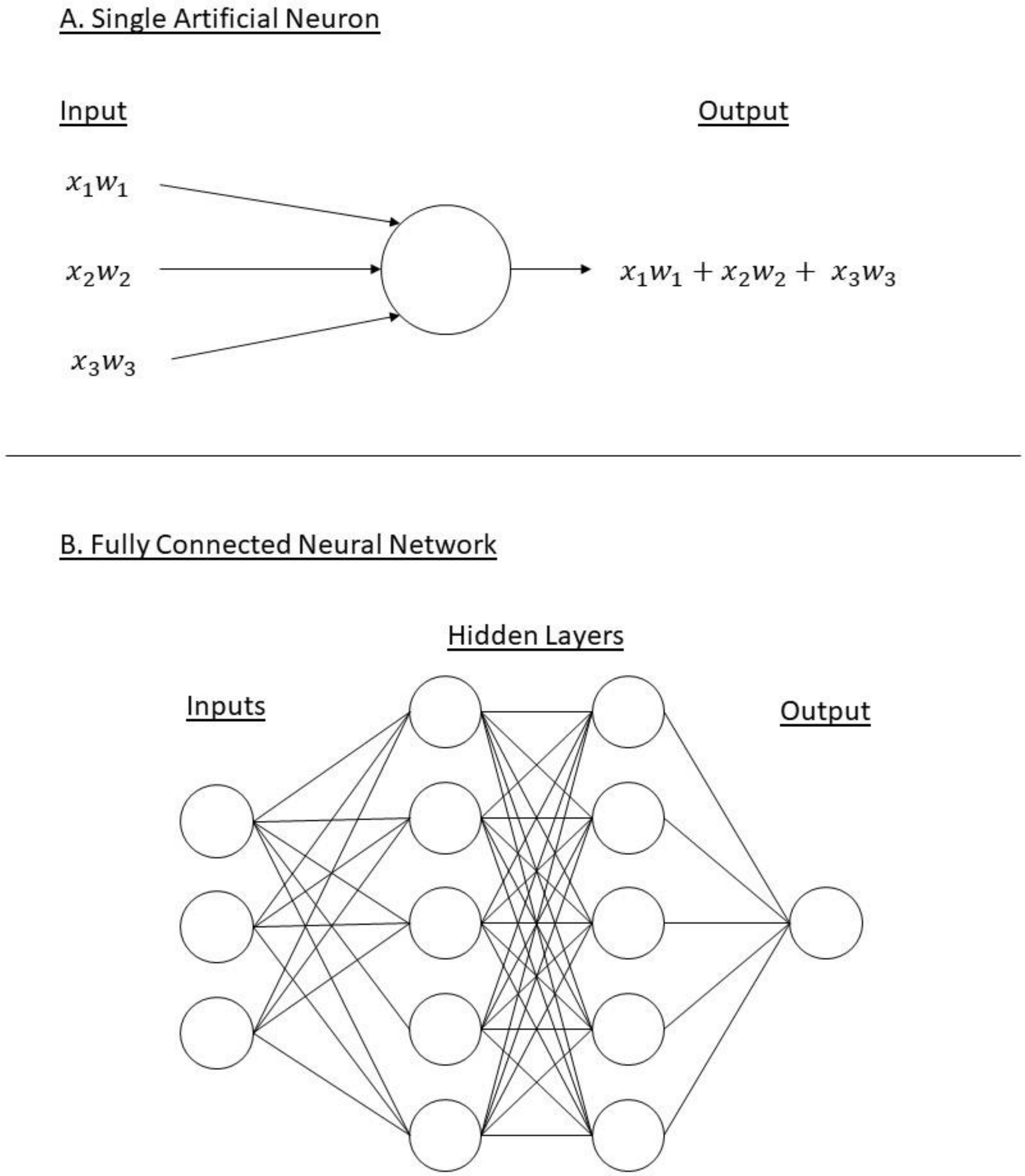

A. Single Artificial Neuron
Input
Output
$x_1w_1$
$x_2w_2$
$x_3w_3$
$x_1w_1 + x_2w_2 + x_3w_3$
B. Fully Connected Neural Network
Hidden Layers
Inputs
Output

Figure 1.3: A.) Schematic representation of a single artificial neuron. Weights, w, are applied to input values, x, via the dot product to produce a scaler output. B.) Multiple artificial neurons are connected to create a complex web of interactions. These complex connections provide this model the ability to learn and solve tasks by adjusting weights at each layer.

# CHAPTER 2

# Adipose Tissue Classification

## Introduction

### *Fat Classification and Chemical-Shift Encoded Magnetic Resonance Imaging*

The role of white adipose tissue (WAT) and brown adipose tissue (BAT) in diseases such as obesity and diabetes has been known for some time [1]. The cells that make up WAT are well suited to storing fat in the form of triglycerides, while the cells that make up BAT, rich in mitochondria and blood vessels, have the ability to consume triglycerides [1]. Since an overabundance of WAT in the body is associated with disease, determining the volume and distribution of WAT and BAT in the body by MRI has increased in clinical importance. Water-fat imaging is a very active field in MRI research and chemical-shift encoded magnetic resonance imaging has been shown to be successful at quantifying the triglyceride composition of adipose tissues [2]. These methods find MR signal contributions attributed to water and fat separately, including three triglyceride composition parameters from which individual spectroscopic peaks can be found using theoretical peak amplitudes.

Previous methods using chemical-shift encoded magnetic resonance imaging to differentiate WAT and BAT using water-fat separation techniques have assumed a fixed-peak spectroscopic model for both types of adipose tissue [3]. These methods only solve for water and fat components of the signal, but not the triglyceride composition parameters. The primary features used for tissue differentiation using these models are T2* relaxation times (or R2*) and fat fraction ratios. However, in reality the amplitudes of spectroscopic peaks vary depending on the type of adipose tissue and its triglyceride composition, and therefore, these fixed-peak models have led to inaccuracies [4].

Work done by Pernilla Peterson and Sven Månsson describes the process behind decomposing a complex gradient echo signal into proton densities in water and fat and triglyceride composition parameters [5]. The complex signal from a voxel that contains both fat and water can be expressed as

$$s(t) = \left(\rho_w + \rho_f \sum_{n=1}^{N} \alpha_n e^{i2\pi f_n t}\right) \cdot e^{i2\pi f_B t} \cdot e^{-R_2^* t} \quad (6)$$

Here $\alpha_n$ represents the relative amplitude of the *n*th fat resonance peak, $f_n$ is the resonant frequency of the *n*th fat resonance peak with respect to water, $\rho_w$ and $\rho_f$ are water and fat proton densities, $f_B$is the local static field inhomogeneity with respect to $B_0$, and $R_2^*$ =1/ $T_2^*$ is the transverse relaxation rate. The chemical structures of fatty acids are well studied, and therefore, *a priori* knowledge of the chemical shift frequencies and relative amplitudes can be used in the decomposition [5]. The relative amplitudes can be expressed as functions of three triglyceride parameters: fatty acid chain length (cl), number of double bonds (ndb), and the number of methylene-interrupted double bonds (nmidb) [2]. **Table 2.1** shows a list of values for both chemical shift frequencies and theoretical amplitudes derived by Hamilton et al [2], [3].

**Equation (6)** can be expressed in matrix form as $\boldsymbol{S}_{Nx1} = \boldsymbol{\Psi A P}$ where,

$$\boldsymbol{P} = \begin{bmatrix} \rho_w \\ \rho_f * F \\ \rho_f * F * ndb \\ \rho_f * F * \ nmidb \\ \rho_f * F * cl \end{bmatrix} \qquad (7)$$

where F is equal to 1/(6*cl – 2*ndb + 2). Solving for $\boldsymbol{P}$ allows for the determination of specific values for ndb, nmidb, and cl that can then be used to solve for the individual peak amplitudes via **Table 2.1**. The individual peak amplitudes, along with features such as T2* and fat fraction can then be used for classification in machine learning algorithms.

## Methods

The success of image classification in machine learning in general is dependent upon the features used to train the models in the datasets [6]. If the values for separate classes are too similar, then it will be difficult for a model to make predictions about an input's identity. Therefore, by providing more information to an algorithm it can be assumed that this should improve classification. While previous work has been done to classify BAT and WAT in mice models utilizing T2* and fat fraction values, using datasets comprised of these values with the addition of features from varying fat-peak spectroscopic models has gone unexplored [7].

In this work four supervised machine learning models are used to classify and differentiate BAT and WAT in MRI images of mice, with features generated from a varying fat-peak spectroscopic decomposition. Three of these models, the fully-connected neural network (FCNN), one-dimensional convolutional neural network (1D CNN) and two-dimensional convolutional neural network (2D CNN), are deep learning models while the fourth machine learning model, the support vector machine (SVM) is a non-probabilistic binary classifier.

SVMs classify data by determining an (n-1)-dimensional separating hyperplane that best separates data, where n is the number of features per sample [8]. Each feature in a dataset represents one dimension in a sample's data space. The hyperplane is chosen by maximizing the margin between support vectors and the hyperplane. Support vectors are data points that are the closest to the separating hyperplane**. Figure 2.1** displays a schematic representation of the hyperplane and support vectors in an example of a two-dimensional dataset. While in their standard form SVMs are best suited for linearly separable data, nonlinearly separable data can be transformed using a radial basis function (RBF) allowing the SVM to separate the data in the transformed space. RBFs are real-valued functions that require that an output value depend only the input values' distance from some defined origin. While RBFs can take many forms, the most usual form used in SVM is the gaussian function of the form,

$$K(x, x') = e^{(-\gamma||x-x\prime||^2)} \qquad (8)$$

where x is a vector of input values, x' is the origin, and $\gamma$ is a tuning parameter specific to the dataset. Applying the RBF to the dataset potentially transforms an originally linearly inseparable dataset into a separable one.

Since most BAT and WAT classification strategies often use T2* and fat fraction, it was also advantageous to see how well each classification model would perform using the raw signal from the MRI [7], [9], [10], [11]. This raw signal comes in the form of 12 complex echo values generated during data acquisition.

### *Data Collection*

MRI data was collected from the inter-capsular BAT (iBAT) and inguinal WAT (igWAT) regions of 2-month-old C57/BL6 mice (n=6) using a 7T Varian Magnex MRI Scanner. All experimental procedures were approved by the Institutional Animal Use and Care Committee (IAUCC, University of Georgia). The mice were anesthetized with isoflurane in the scanner throughout the entire study. The mice were scanned in two separate groups on separate days. Group 1 consisted of four mice and Group 2 consisted of two mice. Both groups used a 2D multi-echo gradient-echo sequence. Group 1 scans consisted of 12 echoes with a spacing of 0.525 ms, an initial echo at 5.7 ms., a FOV of 25 x 25 mm and a resolution of 256 x 256 voxels

The number of slices varied between 2 and 5 depending upon the actual size of the iBAT and igWAT regions. Group 2 scans consisted of 12 echoes with a spacing of 0.525ms with an initial echo at 3.0 ms. There were 5 slices for each region with a FOV of 30 x 30 mm and a resolution of 128 x 128 voxels.

Masks were manually created to consistently extract the same voxels of BAT, WAT and muscle for each of the slices in the data collection. Two datasets were created from the MRI acquisitions. A *Peak dataset* consisting of five spectroscopic peak values derived from a 9-peak model, water proton density, fat proton density, R2* and fat fraction was produced. The five spectroscopic peaks include peaks 1, 4, 6, and 8 in **Table 2.1** that are explicit functions of the triglyceride parameters ndb, nmidb, and cl. The final peak value was calculated from the second element of **Equation 7**, which is essentially the fat proton density divided by the sum of all fat peak values. This was done to include fat peaks that are not explicit functions of the triglyceride parameters ndb, nmidb, and cl. Parameters of the Peak dataset that resulted in infeasible values (e.g., negative values for peak intensities) were removed from each of the datasets. An *Echo dataset* was created using the complex echo signal of the 12 echoes during data acquisition. To prepare the Echo dataset for classification, real and imaginary components from the complex data were organized by alternating real and imaginary components for each echo effectively creating a one-dimensional vector of length 24 for each voxel.

### *Classification Models*

The total number of samples in each of the datasets was 3,258 which consisted of an equal number of samples for each of the three classes: WAT, BAT, and muscle. The datasets were randomly organized into a training set (75%) and validation set (25%) of the total datasets respectively. Each feature in both datasets was standardized to have a mean of zero and a standard deviation of one.

Four different supervised machine learning methodologies were applied to each dataset to test their relative accuracies and total fat classification effectiveness. The four models are

- Support Vector Machine (SVM)
- Fully-Connected Neural Network (FCNN)
- One-dimensional Convolutional Neural Network (1D CNN)
- Two-dimensional Convolution Neural Network (2D CNN)

The input parameters for the first three models were one-dimensional vectors representing the number of features (num_features), and two-dimensional vectors for the 2D CNN.

### *SVM*

The SVM models used a radial basis function kernel with a $\gamma$ of .01 and a penalty parameter of 35.0. Certain SVM algorithms allow outlier data points to lie on the opposite side of the hyperplane. The penalty parameter is a data specific value that is applied to these outliers to control how many data points are allowed to be on the opposite side of the hyperplane. An overall greater accuracy can be achieved by allowing some outliers to be misclassified. A one-against-one approach was used for multi-class classification.

### *FCNN*

The FCNN models (**Figure 2.1A**) consisted of three fully connected layers of 60, 30, and 3 hidden units respectively. Between each of the layers a dropout of 0.25 was implemented as this has been seen to improve the generalization of these types of classifiers [12]. Dropout is the process of temporarily excluding a certain percentage of weights from the training process during a single epoch which has been shown to improve the speed of training as well as aiding in preventing overfitting of a trained model. The outputs of the first two layers utilized a rectified linear unit (ReLU) function and the final output layer used a softmax function. The softmax function is a normalized exponential function of the form,

$$f(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}} \qquad (9)$$

where j is the dimension of the input x, and i is the number of nodes in the output. This function takes each of the outputs and puts them in the range [0,1] such that they add up to 1. Therefore, the node that has the largest number signifies the identity of the input data. The models were trained using a mean squared error loss function and stochastic gradient descent (SGD).

*1D CNN*

The 1D CNN models (**Figure 2.1B**) consisted of two layers of convolution feeding into a final fully connected output with 3 hidden units. Between each of the layers a max pooling layer was implemented. The first layer of convolution had 64 filters of length 6 which then lead to a max pooling of length 4 via a ReLU function. The second layer of convolution had 32 filters of length 4 which then lead to a max pooling of length 2 via a ReLU function. After each max pooling layer, a dropout of 0.25 was used. The final layer of max pooling was flattened and output to the final fully connected layer via a softmax function. The models were trained using a mean squared error loss function and SGD.

*2D CNN*

The inputs for the 2D CNN were obtained differently compared to the other three models. Patches of size 9x9 voxels were obtained for each sample in the dataset. The center voxel for each of the samples corresponded to the same voxels used in the previous models' datasets, with the remaining voxels in the 9x9 grid consisting of voxels directly surrounding the center. The center voxel determined the class of the sample and therefore the total number of samples in the 2D CNN datasets were identical to the other models. The overall shape of the input samples therefore was 9 x 9 x num_features.

The 2D CNN models' architecture (**Figure 2.1C**) consisted of two layers of convolution feeding into a final fully connected output with 3 hidden units. Between each of the layers a max pooling layer was implemented. The first layer of convolution had 64 filters of length 6x6 which then lead to a max pooling via a length of 4x4 via a ReLU function. The second layer of convolution had 32 filters of length 4x4 which then lead to a max pooling of length 2x2 via a ReLU function. After each max pooling layer, a dropout of 0.25 was used. The models were trained using a mean squared error loss function and SGD. The final layer of max pooling was flattened and output to the final fully connected layer via a softmax function.

### Data Analysis

To determine the accuracies of each of the models, ten separate runs were performed for each of the models and then averaged for both the training and validation sets of data, **Table 2.2**. Confusion matrices and ROC curves were generated for each of the models' validation results, **Figure 2.3**. A confusion matrix is a table used to display the performance of a machine learning algorithm. Each row of the table represents the percentage of the dataset that has received a certain prediction and each column represents the actual identity of a voxel. Therefore, the diagonal of the matrix represents the percentages of voxels that have been labeled correctly for each class, whereas the off-diagonal elements represent percentages of incorrectly labeled voxels. By using this matrix, it can easily be seen which classes are more difficult to separate for the model.

An ROC curve, or receiver operating characteristic curve, is a plot that visualizes the accuracy of a model at different confidence thresholds. The outputs of the softmax function can be viewed as a confidence value the classifier has about a certain prediction since the total values of all the outputs add up to 1. In the ROC curves, the vertical axis displays the true positive rate which is the case where an input is correctly classified. The horizontal axis displays the false positive rate which is the case where an input is incorrectly classified.  A diagonal line that can be seen in each of the ROC curve plots represents a line in which the classifier would not be better than simple random guessing. All curves above this line represent classifiers that are better than random chance, and the larger the area under these curves the more accurate a model is considered.

## Discussion

The resulting accuracies of each model and their standard deviations over ten runs are displayed in **Table 2.2**. The lowest testing and validation accuracies when considering each dataset and each model is the SVM using the Peak Dataset with values 95.09% and 94.93% respectively. It is not unexpected that the SVM would present the lowest accuracy of all the models since this type of machine learning model is simply a linear classifier with the least number of tunable parameters. However, even though these values are the lowest over all, they still represent a highly accurate classifier, which suggests that each of the models and their corresponding datasets are sufficient for classification.

The dominant model with the best performance in both training and validation for both datasets is the 2D CNN, with the Echo dataset showing a small improvement over the Peak dataset. These high accuracy values for the 2D CNN are likely due to the addition of spatial information provided by the 2D CNN. With each of the other models, only information about the specific voxel the input comes from is provided, while the 2D CNN uses information from the 9x9 grid surrounding the center voxel.

It can be seen from the confusion matrices and the ROC curve plots in **Figure 2.3** that the major error in class differentiation comes from the difficulty in separating WAT and BAT. In each case, at low confidence values in the ROC curves, false negatives for BAT present the biggest problem. This can be seen by the red curve in each ROC plot. The area under this curve tends to be slightly lower in each case compared to the other two classes. As before, in each model, the Peak dataset shows more difficulty differentiating WAT from BAT compared to the Echo dataset. Since WAT and BAT are much closer anatomically and physiologically to each other than they are to muscle, these results make sense.

Every model presented shows considerably high accuracies and could be used to confidently predict quantities of WAT and BAT, however it is interesting that the Echo dataset is consistently slightly more accurate than the Peak dataset. The Peak dataset contains information extracted from the signal represented by the Echo dataset, so each dataset contains the same relationships albeit in a nonlinear way. Surprisingly even in its raw form the Echo dataset can produce an accurate classifier, possibly due to the increased dimensionality per input for this dataset.

## Conclusion

In this chapter, the comparison of multiple machine learning models have been compared in order to demonstrate the power of machine learning for image segmentation in MRI. While each model proved to be a sufficiently accurate classifier, it was shown that simply using a raw form of the echo signal from the MRI can form a dataset for accurate classification, especially when used as input for a 2D CNN.  This result could decrease classification time for determining quantifies of WAT and BAT in the body, since no peak data calculations need to be performed to provide accurate predictions. To test the full potential and accuracy of these results, histological results should be performed to accurately compare the results of the machine learning algorithms. If successful, this would allow for rapid calculations of WAT and BAT quantities shortly after MRI signal acquisition.

## References

[1] S. Cinti, "Between brown and white: Novel aspects of adipocyte differentiation," *Ann. Med.*, vol. 43, no. 2, pp. 104–115, 2011.

[2] G. Hamilton, T. Yokoo, M. Bydder, I. Cruite, M. E. Schroeder, C. B. Sirlin, and M. S. Middleton, "In vivo characterization of the liver fat $^{1}$H MR spectrum.," *NMR Biomed.*, vol. 24, no. 7, pp. 784–90, 2011.

[3] G. Hamilton, D. L. Smith, M. Bydder, K. S. Nayak, and H. H. Hu, "MR properties of brown and white adipose tissues," *J. Magn. Reson. Imaging*, vol. 34, no. 2, pp. 468–473, 2011.

[4] C. Zancanaro, R. Nano, C. Marchioro, A. Sbarbati, A. Boicelli, and F. Osculati, "Magnetic resonance spectroscopy investigations of brown adipose tissue and isolated

brown adipocytes.," *J. Lipid Res.*, vol. 35, no. 12, pp. 2191–2199, 1994.

[5] P. Peterson and S. Månsson, "Simultaneous quantification of fat content and fatty acid composition using MR imaging," *Magn. Reson. Med.*, vol. 69, no. 3, pp. 688–697, 2013.

[6] A. Halevy, P. Norvig, and F. Pereira, "The Unreasonable Effectiveness of Data," *IEEE Intell. Syst.*, vol. 24, no. 2, pp. 8–12, 2009.

[7] K. N. Bhanu Prakash, S. K. Verma, J. Yaligar, J. Goggi, V. Gopalan, S. S. Lee, X. Tian, S. Sugii, M. K. S. Leow, K. Bhakoo, and S. S. Velan, "Segmentation and characterization of interscapular brown adipose tissue in rats by multi-parametric magnetic resonance imaging," *Magn. Reson. Mater. Physics, Biol. Med.*, vol. 29, no. 2, pp. 277–286, 2016.

[8] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, 1992, pp. 144–152.

[9] E. Lunati, P. Marzola, E. Nicolato, M. Fedrigo, M. Villa, and a Sbarbati, "In vivo quantitative lipidic map of brown adipose tissue by chemical shift imaging at 4.7 Tesla.," *J. Lipid Res.*, vol. 40, no. 8, pp. 1395–1400, 1999.

[10] S. B. Reeder, H. H. Hu, and C. B. Sirlin, "Proton density fat-fraction: A standardized mr-based biomarker of tissue fat concentration," *Journal of Magnetic Resonance Imaging*, vol. 36, no. 5. pp. 1011–1014, 2012.

[11] K. N. Bhanu Prakash, H. Srour, S. S. Velan, and K.-H. Chuang, "A method for the automatic segmentation of brown adipose tissue," *Magn. Reson. Mater. Physics, Biol. Med.*, vol. 29, no. 2, pp. 287–299, 2016.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Mach. Learn. Res.*, vol. 15,

pp. 1929–1958, 2014.

Table 2.1: Theoretical amplitudes of each fat resonance peak for a theoretical 9 fat peak model given in terms of the number of double bonds (ndb), number of methylene-interrupted double bonds per molecule (nmidb), and the fatty acid chain length (cl). Chemical shift values in ppm are provided for each of the theoretical peak amplitudes.

**Chemical Shifts and Theoretical Amplitudes Relative to Water** [2], [3]

| Peak Number | Chemical Shift (ppm) | Theoretical Peak Amplitude |
|---|---|---|
| 1 | 0.59 | 2ndb |
| 2 | 0.49 | 1 |
| 3 | -0.50 | 4 |
| 4 | -1.95 | 2nmidb |
| 5 | -2.46 | 6 |
| 6 | -2.68 | 4(ndb – nmidb) |
| 7 | -3.10 | 6 |
| 8 | -3.40 | 6*(cl-4)-8*ndb+2*nmidb |
| 9 | -3.80 | 9 |

Table 2.2: Model accuracies and standard deviations for each of the machine learning models and each dataset. Training accuracies describe the accuracies of the training dataset used during the training process and upon which SGD is used to adjust model weights. Validation accuracies describe the accuracies of the validation, or testing, dataset.

**Model accuracy percentages and standard deviations**

| Model | Peak Dataset | | | | Echo Dataset | | | |
|---|---|---|---|---|---|---|---|---|
| | SVM | FCNN | 1D CNN | 2D CNN | SVM | FCNN | 1D CNN | 2D CNN |
| Training Accuracy | 95.09<br>(0.36) | 95.17<br>(0.20) | 95.43<br>(0.38) | 97.58<br>(0.17) | 97.41<br>(0.23) | 96.45<br>(0.49) | 97.37<br>(0.37) | 99.17<br>(0.18) |
| Validation Accuracy | 94.93<br>(0.71) | 94.72<br>(0.82) | 94.51<br>(0.63) | 96.70<br>(0.44) | 96.28<br>(0.45) | 95.05<br>(0.77) | 96.20<br>(0.58) | 98.43<br>(0.50) |

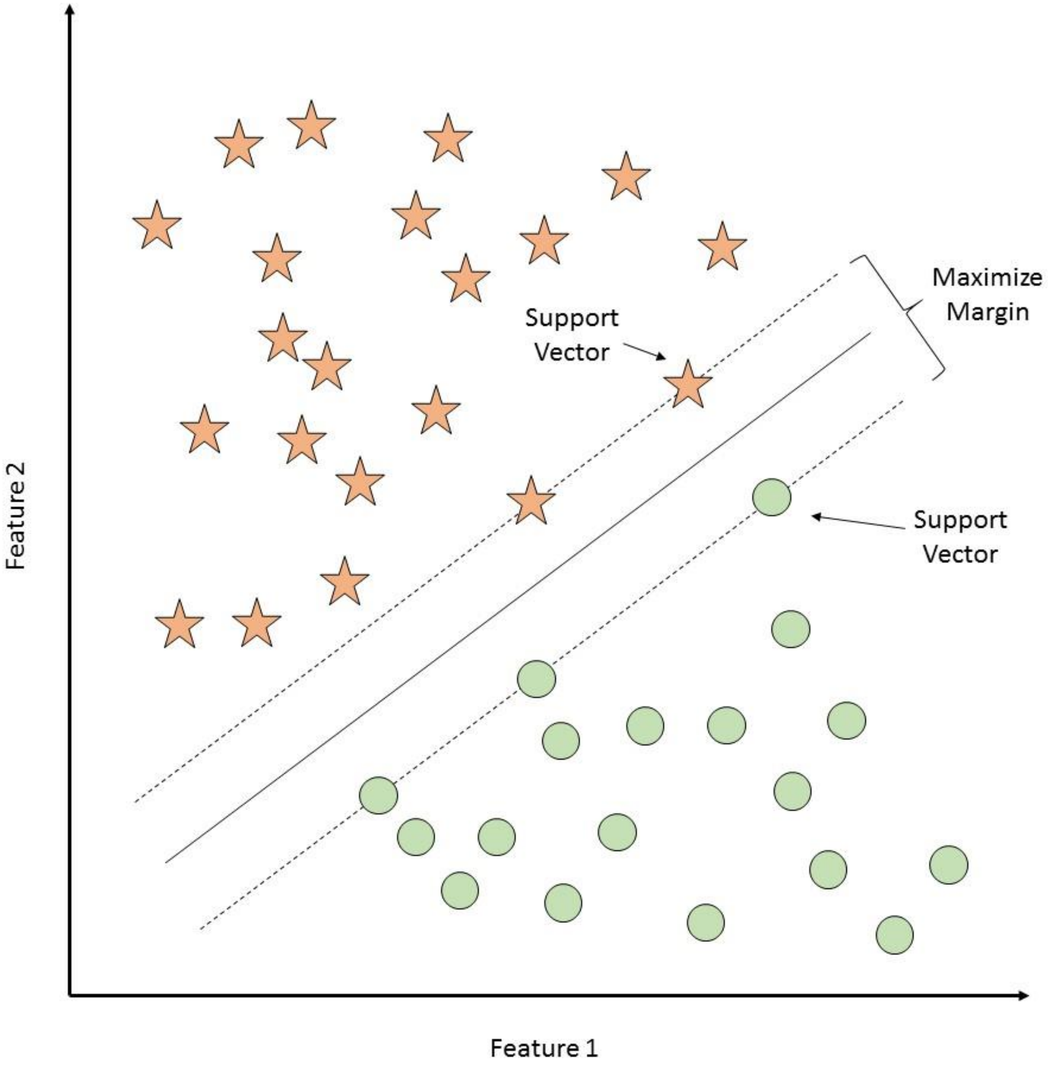


Figure 2.1: Schematic representation of the SVM algorithm. Data points comprised of two features are plotted along with the separating hyperplane that best maximizes the margin distance between support vectors. Support vectors are data points that lie along lines formed a distance equal to half the margin from the hyperplane. Data points on either side of the hyperplane belong to separate classes.

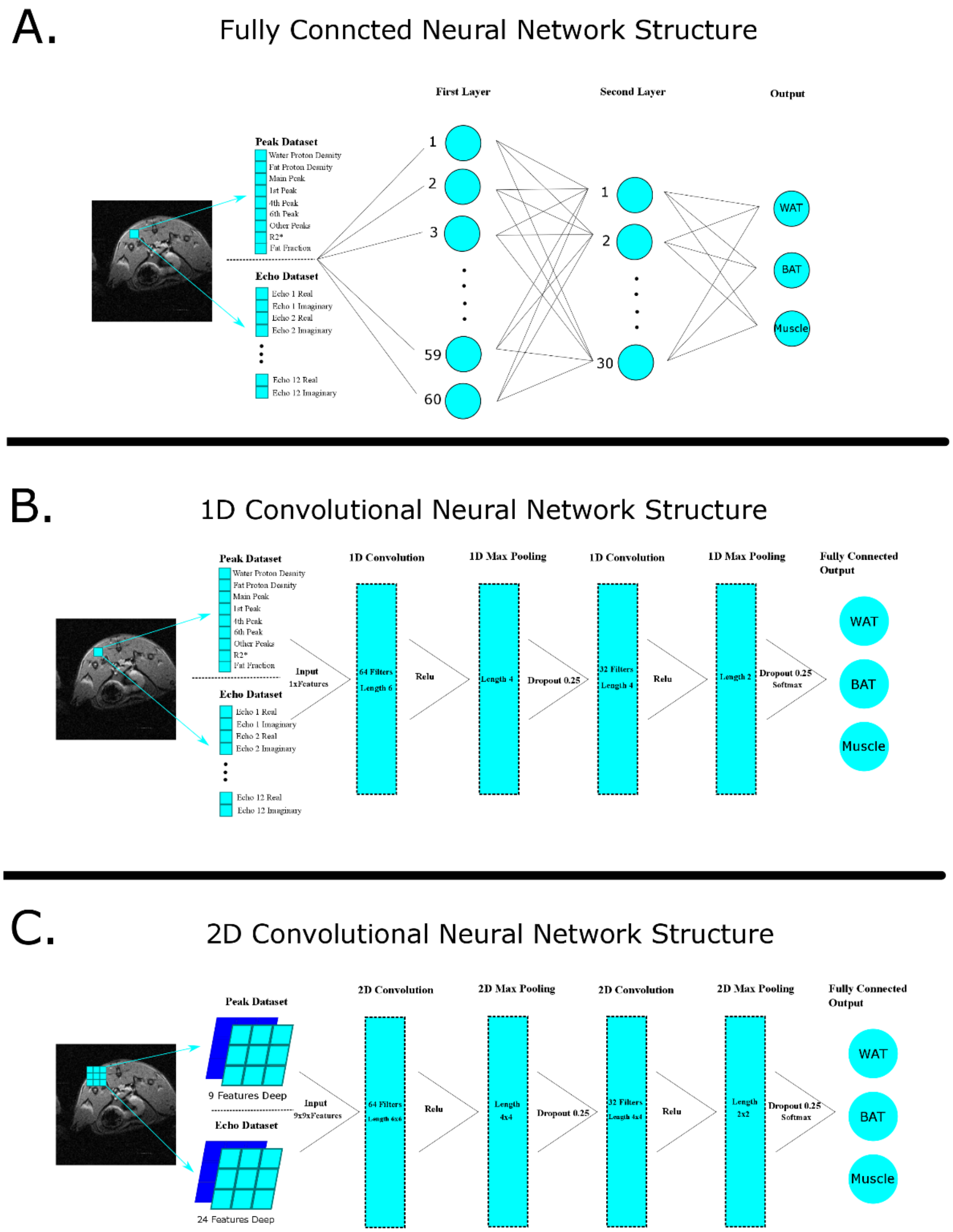


Figure 2.2: Schematic representation of A) the input data structures and model architecture of the FCNN, B) the input data strctures and model architecture of the 1D-CNN, and C)  the input data structures and model architecture of the 2D CNN.

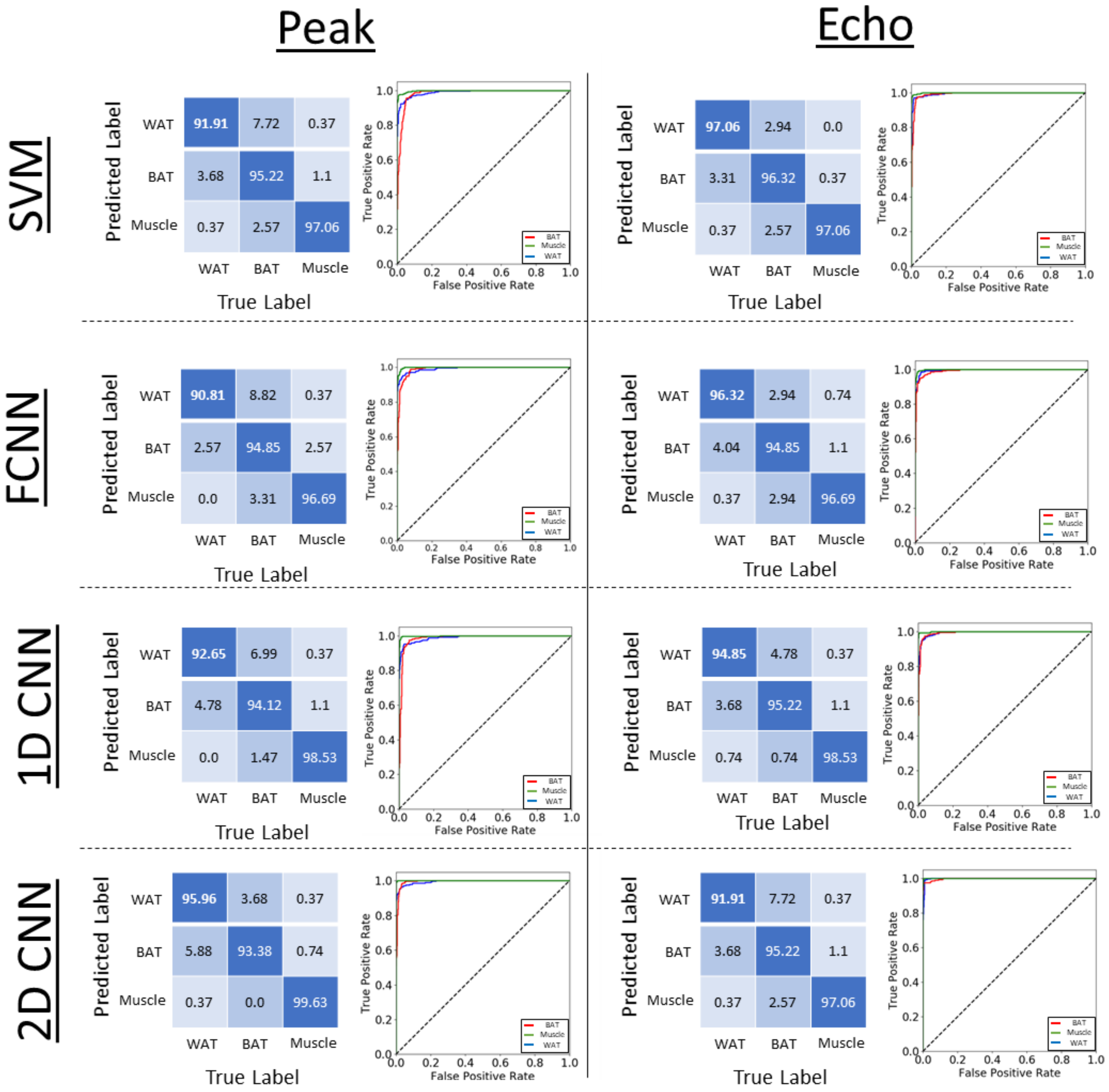


Figure 2.3: Confusion matrices and ROC curves for each of the models and datasets. The confusion matrices show how well each model makes predictions on each class. The rows of the matrices are predicted labels from the machine learning algorithm, while columns are the true labels of the data. Each row adds up to 100% to show in class accuracy. ROC curves illustrate how well a model makes predictions on each class at different confidence thresholds. The blue curve represents WAT, the red curve is BAT and the green curve is muscle.

# CHAPTER 3

# Brain Functional Connectivity Analysis via a CNN Sparse Autoencoder

## Introduction

### *Brain Functional and Structural Analysis*

Brains function as a very complex network of communications between a large number of individual cortical and subcortical regions in the brain [1]–[3]. While each of these regions may have their own intrinsic function, certain areas of the brain still interact even when there is no obvious task being performed. The networks that comprise these interactions are referred to as resting state networks (RSNs) and are often identified via changes in blood flow, electrophysiological changes measured by electroencephalography (EEG), or magnetoencephalography (MEG) in the brain during states of rest in a subject. In recent years functional Magnetic Resonance Imaging (fMRI) has been used to uncover these resting state networks by measuring blood-oxygen-level dependent (BOLD) signals.

BOLD imaging works based on the body's hemodynamic response [4]. A hemodynamic response occurs in neuronal tissue in response to cellular demands for oxygen and glucose. Neurons do not possess the ability to store these nutrients internally, and therefore when specific neurons fire they require that these nutrients be supplied via oxygenated blood. The iron-containing carrier of oxygen in blood is hemoglobin [4]. The oxygenated and deoxygenated forms of hemoglobin, respectively oxyhemoglobin and deoxyhemoglobin, possess different magnetic susceptibilities and therefore their relative levels can be detected via fMRI signal magnitudes. A larger amount of oxyhemoglobin in an area of the brain suggests that the neurons in this area are being activated and is represented as an increased MRI signal from these locations for the duration of time that these regions are active [4].

Time-varying series of BOLD signals are acquired via an echo planar imaging (EPI) sequence. This sequence allows for the collection of signals from an entire brain volume in a single RF excitation allowing brain states to be collected during a certain period of time (e.g. 20 minutes). RSNs can then be determined from the collected time-series brain signals over a specific MRI signal acquisition session. One of the most common ways to uncover RSNs is using seed based analysis on the fMRI time-series signals [5]. Seed based analysis attempts to find correlations between a selected seed region compared to the rest of the brain. The seed region is usually chosen as an averaged time-series over a collection of voxels in a region of the brain suspected or known to be active during rest. Using this averaged time-series seed, it is then compared pairwise with each other voxel in the brain where similarity measures can be used to determine how similar or connected the voxels are to the seed region. Connected regions tend to show similar fluctuations in signal magnitude or frequency.

Since this seed based approach requires *a prior* information it is most suited to be used on human subjects where there is a lot of historical information about the connectivity of the brain. However, for other subject species this information is lacking and therefore other approaches may be useful. Pig brain models have become important due to their shared similarities to humans compared to other animal models, especially due to the increased interest and research devoted to degenerative brain disorders (e.g. Alzheimer’s Disease), stroke, traumatic brain injuries, etc. [6]–[8]. Comparing the structural differences between healthy pig brains and trauma induced pig brains has proven to be a good surrogate for understanding how trauma affects the human brain, but further work must be done to understand how functional connectivity changes between healthy and trauma induced pig brains. During acquisition of fMRI images, pigs must be anesthetized to reduce motion artifacts. This creates the need for discovering RSNs in the fMRI signal of the pig to understand functional connectivity in healthy and trauma induced pigs. Since much is unknown about the connectivity of the pig brain a new method beyond seed based analysis is needed to uncover these RSNs. In this chapter, we discuss a new method to find RSNs from fMRI time-series data based on a CNN sparse autoencoder algorithm [9]. In addition to functional connectivity, the process of determining structural connectivity in the brain through diffusion weighted imaging (DWI) and diffusion tenor imaging (DTI) is discussed. DWI images track the Brownian motion of water molecules in an image voxel allowing for the overall diffusion of water to be tracked throughout the brain [10]. Within the brain the flow of water generally travels along neuronal connections following the structural connectivity of the brain. Understanding both the functional and structural connectivity in the brain is necessary to understand how the brain works.

## Methods

### *Sparse Autoencoder*

Autoencoders are an unsupervised learning algorithm that can learn key features from unlabeled datasets [9], [11]. An autoencoder algorithm can utilize diverse types of deep learning layers, such as FCNNs or CNNs, but the overall goal of the autoencoder is to approximate an identity function of the input data while storing valuable information about the dataset in its weight values. The total number of weights that connect different layers in any given model is limited, therefore for the model to be able to reliably recreate any given input in a dataset, restrictions must be imposed so that only common, repeated features in a dataset be recorded in the weights [9], [11]. If a dataset was completely random then this learning task would prove to be impossible. As a relevant example, consider a time-series dataset with signals of length 50 as an input into a three-layer FCNN with 25 hidden units in its middle layer and 50 hidden units in its output. Essentially, the input layer of 50 must be transformed to a length of 25 in the middle layer and then back to the original length of 50 in the output layer. For the autoencoder to be successful, this dimensionality reduction between the input and middle layer of the FCNN requires that the weights be chosen in such a way that only the most key features of the dataset be chosen to achieve the lowest loss value possible. Since a completely identical reconstruction of any given signal would be impossible, reconstructions from autoencoders are often described as denoised representations since only the most common features are output from any given signal. In general, these models are trained via SGD and common loss functions such as MSE.

Sparse autoencoders instead increase the dimensionality of a sample and then again reduce the expanded dimensions back to the original sample dimensionality [9]. Again, imagine a three-layer FCNN with the same time-series dataset. The middle layer of this model now contains 500 hidden units instead of 25. Now for any given input sample, many of the hidden units in the middle would not need to be activated to recreate the sample; a sparsity in the activations is created. While the output time series still will not be completely identical, this sparsity allows features to be extracted from inputs. Similar inputs will tend to activate the same hidden nodes and therefore can be grouped together. CNN sparse autoencoders take this idea an extra step with the addition of convolutional kernels and will form the basis for RSN analysis presented in this chapter. Due to the lack of *a priori* information about the RSN connectivity of the pig brain a new method based on a CNN sparse autoencoder (CNN-SA) was explored in this work.

### *Diffusion Weighted Imaging and Fractional Anisotropy*

DWI is an MRI technique that measures the diffusion of water in various tissues by tracking the Brownian motion of water molecules [10]. This type of imaging is particularly useful for understanding the structural connectivity of the brain. Neuronal cells that conduct the electrical impulses that control the brain possess long tracts called axons. The long cylindrical axons are covered in cell membranes that force water to move anisotropically along their length. By measuring the anisotropically directed movement in the axons compared to other locations that may allow water to move isotropically, the structural connectivity of these axons within the brain can be visualized.

To create diffusion weighted images, directional magnetic gradients are applied to a volume of tissue after the application of a RF excitation pulse [10]. The directional gradient dephases the magnetic moments of each hydrogen atom in a voxel and then the directional gradient is reversed in order to rephase the magnetic moments. In voxels where water is diffusing isotopically, the rephasing of the magnetic moments will not be perfect and therefore the resulting signal in these voxels will be lowered. Voxels that have lower rates of diffusion will be rephased and have higher signals. By applying these gradients in multiple directions, the directional flow of water can be determined.

If gradients are applied in at least six separate directions a 3x3 diffusion tensor can be calculated for each voxel that describes the anisotropy of diffusion in that voxel [12]. The main eigenvector of this tensor is associated with the direction of the fibers that connect in the brain structurally. Fractional anisotropy (FA) is an extension of diffusion tensor imaging that tracts the fibrous connections between each voxel in order to determine individual fibers and connections throughout the brain [12]. FA utilizes eigenvalues from the diffusion tensor to produce a scaler value that determines the degree of anisotropy in diffusion. For each voxel a value between 0 and 1 is calculated by the equation

$$FA = \sqrt{\frac{1}{2}}\frac{\sqrt{(\lambda_1-\lambda_2)^2+(\lambda_2-\lambda_3)^2+(\lambda_3-\lambda_1)^2}}{\sqrt{\lambda_1^2+\lambda_2^2+\lambda_3^2}} \qquad (10)$$

where $\lambda_1, \lambda_2, and\ \lambda_3$are eigenvalues from the diffusion tensor. A value of 0 describes a voxel with isotropic diffusion and a value of 1 describes a voxel of relative maximal diffusion along the main eigenvector of the voxel's diffusion tensor [12].

Tractography algorithms often use the diffusion tensor and FA results to determine the extent of connections throughout the brain. By applying threshold limitations to the results of FA and the diffusive directions of each voxel, the brain can be iterated over, and structural connections can be determined. There are many different software implementations of tractography using diffusion weighted imaging scans in conjunction with FA that can produce visualizations of the structural connectivity in the brain.

### *Data Acquisition and CNN-SA Structure*

BOLD imaging data was collecting using a 3T GE MRI Scanner from an adult male castrated Landrace pig. To minimize signal changes due to motion, the pig was sedated through an intramuscular injection of xylazine (7 mg/kg), butorphanol (0.3 mg/kg), and midazolam (0.3 mg/kg). Anesthesia was maintained with 1.5% inhalational isoflurane in oxygen. All experimental procedures were approved by the Institutional Animal Use and Care Committee (IAUCC, University of Georgia). The scan consisted of a total of 300 time points with an echo time of 30ms, TR of 2000ms, over 21 slices with a resolution of 1.56 mm x 1.56 mm, and an acquisition matrix of 128 x 128 voxels. The fMRI signals were then preprocessed using the Statistical Parametric Mapping (SPM) software to correct for slice-timing and motion correction during the BOLD scan. A T2-weighted anatomical scan was acquired with a TR of 1376 ms, over 20 slices with a resolution of 1.56 mm x 1.56 mm, and an acquisition matrix of 128 x 128 voxels.  The fMRI data was co-registered to the anatomical data by using SPM to help with determining spatial orientation of potential clusters found in the data.

A DWI scan was performed during the same session as the fMRI and anatomical scans. The scan was performed over 29 slices with a TR of 10,000ms, with an acquisition matrix of 256 x 256. Diffusion tensor calculations, FA and tractography was performed and visualized using the Medinria medical image navigation software [13].

To create the CNN-SA it was necessary to increase the dimensionality of the representation of each signal in the dataset at each layer. **Figure 3.1** shows the overall structure of the CNN-SA. The first layer of the CNN-SA contained 32 kernels each of a length 20 and a stride of one. Each kernel was convoluted over a signal in such a way that it returned 300 new outputs for each kernel which is identical in size to the original time-series. This necessitated zero-padding at the ends of the time signal to fit each kernel over an entire time-series. Therefore, the output for each kernel represents an activation map of how strongly the kernel weights are represented at each convolution stride. These activation maps will be referred to as *correlation maps* in order to remove any association with brain region activations, a concept used in functional tasked-based fMRI analysis.

Since each kernel produces an individual correlation map, the overall representation after this first layer of convolution is 300x32 or 9600 new data points. The second layer of the CNN-SA is a max-pooling layer of two which effectively reduces this new representation down to 4800 points. The following six layers as seen in **Figure 3.1** continue to sparsify the representation of each time-series. These first eight layers of the CNN-SA are referred to as the encoder, or encoding layers. After the eighth-layer, the CNN-SA then reverses the dimensionality of the encoder, mirroring the convolutions by decoding the middle representation of the time-series back into the original 300 points in the time-series.

To train the CNN-SA a mask was created to extract only the brain regions of the fMRI signal data. A total of 16,485 voxels were extracted based on the mask. Training and testing sets were formed from this dataset (16,485 time-series) by randomly selecting 70% of the time-series data for the training set and 30% for testing. The CNN-SA was trained for 50 epochs using an MSE loss function and SGD. **Figure 3.2A** shows the testing and training loss for the CNN-SA over the 50 epochs. It can be seen in this figure that the training and testing losses for the CNN-SA stabilizes by the end of training. **Figure 3.2B** shows examples of original time-series inputs with the outputs of the CNN-SA overlaid in gold. While the reconstructed time-series are not perfect, it is easily seen that the overall structure of the time-series is retained by the reconstruction.

### Analysis and Results

Each successive layer of convolution in the CNN-SA becomes more and more complex and therefore the weights of each kernel in the deeper layers cannot be directly translated to features found in the original time-series. However, since the first-layer of convolution in the CNN-SA has only undergone one layer of convolution, the kernel weights in this layer can directly be compared to the original time-series. Essentially, the weights derived from this layer will represent common shapes represented in the overall time-series dataset.  For this reason, the analysis focused on extracting interesting features and associations between each time-series based on this layer.

The first goal of analysis was to visualize clusters in the data that represent potential RSNs. The first layer of the CNN-SA produced 32 separate correlation maps of length 300 based on the kernel weights in the convolution layer. Since each of these correlation maps plot the strength of each of the kernel weights present at a specific position, these correlation maps can be viewed as how correlated the individual weights are to the input time-series at a specific time. **Figure 3.3** shows plots of each of the 32 kernel weights for the first layer of the CNN-SA. Each kernel in **Figure 3.3** contains a sequential label, 1-32, in the lower left corner for identification. While upon initial inspection these kernel weights may seem random, upon closer inspection certain kernels prove to show interesting trends. For instance, kernel 8 and 18 both show a sinusoidal trend, while kernels 2 and 21 both show a general increase over time. This rising and falling in magnitude is suggestive of the hemodynamic response present in fMRI time-series data. Large values present in correlation maps will suggest that these features are being expressed while lower values suggest the opposite.

To inspect how each kernel is expressed throughout each individual time-series, every correlation map was first scaled so that the values were between 0 and 1. A threshold of 0.8 was applied to each of the correlations and then values of all the correlations above 0.8 were summed. This process was performed for each time-series in the dataset so that each input was associated with a single valued correlation strength for each kernel. These voxels were then mapped back to their original spatial locations to determine if locations in the brain seemed to have clusters of similar high correlation strength to create *correlation strength maps*. Overall 32 separate correlation strength maps were created, one for each kernel.

A second similar form of analysis was performed to see how each kernel is expressed in different sections of time. Again, every correlation map was first scaled so that the values were between 0 and 1. Then each map was divided into 6 subsections of 50 time points each. Once again, a threshold of 0.8 was applied and then values in each subsection of the time series was summed. This resulted in six separate values representing the overall strength of each kernel in a specific subsection of time in each voxel. These voxels were then mapped back to their original spatial locations to create *temporal correlation strength maps*.

The purpose of performing these types of analyses was to determine if identifiable clusters of strong correlation are present throughout the brain. If many similar isolated regions in the brain are correlated to a specific kernel, then they all potentially belong to an RSN. However, since little work has been done to discover RSNs in pigs, a way to determine the legitimacy of the resulting clusters found by the CNN-SA was needed. Therefore, a second type of unsupervised RSN analysis was used known as spectral clustering. Spectral clustering has shown success in discovering RSNs in human brains without the need for *a priori* information and therefore made a good test for the quality of the CNN-SA results. Spectral clustering is a graph-clustering technique that determines the eigenvalues of a similarity matrix, typically an RBF similarity, to perform a dimensionality reduction upon the dataset. Then using these eigenvalues, a traditional clustering method, such as k-means, is applied to complete the clustering of the data. The benefit of using spectral clustering instead of simply k-means is that the data is first transformed to a new space using the RBF function where higher order relationships may be determined. K-means alone often simply clusters values based on distance. Spectral clustering was performed upon the dataset using an implementation of the algorithm in the Scikit-learn package [14].

Lastly, tractography analysis was performed on the DWI scan to visualize the structural connections present throughout the brain using the software Medinria [13].

## Discussion

This process of finding overall correlation strengths for a specific kernel throughout the brain is similar to finding pairwise correlations between an averaged seed time-series and time-series from the rest of the brain. The similarity measures between the seed and other time-series is analogous to using convolutions of each kernel over each time-series. However, the difference here is that this process contains no *a priori* information and is entirely unsupervised. Using the seed based approach, different RSNs are determined by extracting seeds from distinct locations and then performing correlations for each seed, but using the CNN-SA method potentially allows for extracting kernel weights that correspond to features that represent themselves only in specific RSN networks.

Due to the inherent unsupervised nature of the CNN-SA algorithm it is expected that many of the kernel weights will not display any distinct feature present in the overall time-series data, at least in any linear way that can be visualized directly with the original time-series. However, given that the CNN-SA did produce accurate decoded outputs of the original time-series data it must be the case that some of the kernel weights represent valuable information contained throughout the data set.

Upon inspection of the correlation strength maps for each of the 32 kernels, it is apparent that while many of the kernel correlations show very cluttered results without any clear clustering, there are two main types of clusters that are represented consistently through many of the correlation strength maps. **Figure 3.4** illustrates two unique clusterings found consistently throughout the correlation strength maps. While the correlation strength maps were generated over the entire brain, **Figure 3.4** only displays the six most interesting slices for these clusterings. Additionally, beside the correlation strength maps, the kernel weights that produced these maps are displayed as well as a table of Pearson correlation coefficients between these kernels. Each of the kernel weights presented produced correlation strength maps that proved to show spatially consistent correlation locations. Interestingly, but not unexpected, it seems that these kernel weights possess similarities between each other for each cluster.

The kernel weights for *Cluster 1* seem to retain a sinusoidal relationship. The Pearson coefficients between each of the kernels for this cluster show linear correlations. Even though the Pearson coefficients between kernel 8 and the other two kernels are negative, their magnitude is quite large suggesting a strong linear relationship over time. Looking at the actual shape of each of the kernels, it appears that, if viewed as sinusoids, then kernel 8 is 90 degrees out of phase with kernels 18 and 27, explaining the negative correlation. Since the convolution operation is performed step-wise over the entire input time-series the correlation maps would be invariant to the phase of the kernels. The kernel weights for *Cluster 2* similarly seem to possess a shape similar to each other and their Pearson coefficients also show linear correlations. Since each cluster shows consistency in weights, this suggests that these kernel weights are representations of features expressed commonly in the time series dataset.

Looking at the spatial locations of clusters in each of the slices also shows interesting results. The purple, red and yellow colors overlaid in the slices represent voxels where correlation strength of the kernel weights produce a correlation above the threshold of 0.8. Simply the sparsity in locations of high correlation suggest that the kernels are representing a unique feature only represented in unique locations in the brain. Additionally, both *Cluster 1* and *Cluster 2* show symmetry between the two hemispheres of the brain, for instance in the back of the brain for Slice 4 and Slice 5 for *Cluster 2*.

**Figure 3.5** shows results provided by the spectral clustering analysis that most closely match clusters found in **Figure 3.4**. When comparing the two methods it seems that the spectral clustering results are a combination of the clusters found by the CNN-SA method. For instance, in the spectral clustering plot for Slice 1 of **Figure 3.5** the front of the brain shows a large region circled in green that is likewise present in Slice 1 of *Cluster 2*. The spectral clustering for Slice 3 in **Figure 3.5** shows two regions circled in green that are also present in Slice 3 of *Cluster 1*. Other slices show additional similarities. The locations and similar shapes of these similarities suggest that both methods are finding similar regions in the brain that correlate with each other. While there are dissimilarities, the specificity of shape and location in the similarities provide confidence that the CNN-SA is finding unique correlations through analysis of its kernel correlations.

In **Figure 3.6** and **Figure 3.7** the temporal correlation strength maps for both *Cluster 1* and *Cluster 2* are displayed respectively. The top of each figure shows the correlation strength maps from **Figure 3.4** with the corresponding temporal correlation strength maps for each slice in the cluster displayed underneath. The CNN-SA is unique in that it can provide this extra temporal dimension for analysis compared to spectral clustering. Through these temporal correlation strength maps, potential for understanding the temporal structure of connectivity in the brain is possible. In **Figure 3.7** the 51-100 subsection of time clearly shows that the front, middle part of the brain is highly correlated and as time continues this correlation switches to the back of the brain for subsections 201-250 and 251-300. The correlations in **Figure 3.6** also show temporal differences albeit less significantly. In each slice in **Figure 3.6** most of correlations are located in the middle edges of the brain during the 1-50 subsection, then a period of low or no correlation, and then again correlations return to the middle edges of the brain during the 201-250 subsection. This suggests a periodicity in correlation for this cluster type.

The structural connectivity results from DWI analysis are presented in **Figure 3.8**. Each row in the figure shows three separate regional displays of the resulting fiber connections from the tractography algorithm in the Medinria program. The first row displays connections spanning the full pig brain, while the second and third rows display connections separated by subcortical and cortical regions. Each column in **Figure 3.8** displays three separate views of the brain: coronal, sagittal and axial views of the brain.

In the first row, fibers are colored by FA calculations for each voxel in the brain. Yellow and bright green colors correspond to FA values closer to 1 while dark blue corresponds to values closer to 0. Values closer to 1 describe diffusive flows that more strongly follow the direction of the main eigenvector of the diffusion tensor. The color representations in the second and third rows have no inherent meaning beyond making fiber regions more easily visible.

Upon inspection of the structural connections present in **Figure 3.8** it is seen that there are two clear divisions between connections belonging to cortical and subcortical areas. The subcortical region of the mammalian brain is associated with motor movements and homeostasis as well as possessing connections that connect the two hemispheres of the brain [15], [16]. This region receives connections from the spinal cord, and therefore requires fibers that connect many regions of the brain. The cortical region in the brain consists of the outer portion of the mammalian brain. This region contains areas that have roles in complex brain activity such as vision, memory, consciousness and perception [15], [16].

The second row of **Figure 3.8** shows the subcortical structural connections in the brain. Long, continuous fibers that help form connections throughout the brain can be seen. When viewed from the top view, a bridge between the left and right hemispheres of the brain can also be seen. The third row of **Figure 3.8** shows the cortical connections in the brain. The fibers present in this area of the brain are much shorter compared to the subcortical region. Additionally, these fibers have a much more complex trajectory suggesting that many different areas in the cortical tissue of the brain need to be connected. The length and trajectories of the fibers in each of these regions correspond to their overall function.

## Conclusion

In this chapter a new method for determining potential RSN networks in pig brains via an unsupervised CNN-SA has been presented. While further work and verification of these resulting RSNs must be performed, the overall method has shown to produce useful results. To further verify these results, multiple test subjects will need to be analyzed using the same methods where statistical consistency will help determine how effective the method truly will be for finding RSNs. Increasing the amount of test subjects will also provide a larger dataset in which the CNN-SA will be able to refine the kernel weights and further improve the generalizability of the algorithm between separate subjects. Additionally, using clusters found using this method as a type of *a prior* information for the seed based approach of determining RSNs could help verify the accuracy of the CNN-SA and give further evidence of RSNs in the pig brain as well as further comparing these results to spectral clustering analysis.

Once functional connectivity can be supported by further evidence, additional associations between the functional and structural connections in the brain can formed by the methods presented in this chapter. Often there are links between functional and structural connectivity. While these results have not been seen in the structural and functional analysis pretended here, it will be beneficial to continue to look for connections between the two methods. By combining these two types of analysis a concrete understanding of the pig brain can be formed leading to the ability to compare differences between healthy pigs and those with trauma.

## References

[1] M. P. van den Heuvel and H. E. Hulshoff Pol, “Exploring the brain network: A review on resting-state fMRI functional connectivity,” *European Neuropsychopharmacology*, vol. 20, no. 8. pp. 519–534, 2010.

[2] B. B. Biswal, “Resting state fMRI: A personal history,” *NeuroImage*, vol. 62, no. 2. pp. 938–944, 2012.

[3] R. L. Buckner, F. M. Hrienen, and T. B. T. Yeo, “Opportunities and limitations of intrinsic functional connectivity MRI,” *Nat. Rev. Neurosci.*, vol. 16, no. 7, pp. 832–837, 2013.

[4] S. Ogawa, T. M. Lee, A. R. Kay, and D. W. Tank, “Brain magnetic resonance imaging with contrast dependent on blood oxygenation.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 87, no. 24, pp. 9868–72, 1990.

[5] R. L. Buckner, J. R. Andrews-Hanna, and D. L. Schacter, “The Brain’s Default Network,” *Ann. N. Y. Acad. Sci.*, vol. 1124, no. 1, pp. 1–38, 2008.

[6] W. R. Douglas, “OF PIGS AND MEN AND RESEARCH,” *Space Life Sci.*, vol. 3, pp. 226–234, 1972.

[7] K. Gutierrez, N. Dicks, W. G. Glanzner, L. B. Agellon, and V. Bordignon, “Efficacy of the porcine species in biomedical research,” *Front. Genet.*, vol. 6, no. SEP, 2015.

[8] N. M. Lind, A. Moustgaard, J. Jelsing, G. Vajta, P. Cumming, and A. K. Hansen, "The use of pigs in neuroscience: Modeling brain disorders," *Neuroscience and Biobehavioral Reviews*, vol. 31, no. 5. pp. 728–751, 2007.

[9] A. Ng, "Sparse autoencoder." CS294A Lectures Notes 72, pp. 1–19, 2011.

[10] R. Bammer, "Basic principles of diffusion-weighted imaging," *Eur. J. Radiol.*, vol. 45, no. 3, pp. 169–184, 2003.

[11] A. Goodfellow, Ian, Bengio, Yoshua, Courville, "Deep Learning," *MIT Press*, 2016. [Online]. Available: http://www.deeplearningbook.org/.

[12] D. Le Bihan, J. F. Mangin, C. Poupon, C. a Clark, S. Pappata, N. Molko, and H. Chabriat, "Diffusion tensor imaging: concepts and applications.," *J. Magn. Reson. Imaging*, vol. 13, no. 4, pp. 534–546, 2001.

[13] P. Fillard and N. Toussaint, "Medical Image Navigation and Research Tool by INRIA ( MedINRIA )," *Enseignement*, no. November, pp. 1–39, 2006.

[14] F. Pedregosa and G. Varoquaux, *Scikit-learn: Machine learning in Python*, vol. 12. 2011.

[15] R. Olivares, J. Montiel, and F. Aboitiz, "Species differences and similarities in the fine structure of the mammalian corpus callosum.," *Brain. Behav. Evol.*, vol. 57, no. 2, pp. 98–105, 2001.

[16] V. Schmidt, "Comparative anatomy of the pig brain," Justus-Liebig-Universität Gießen, 2015.

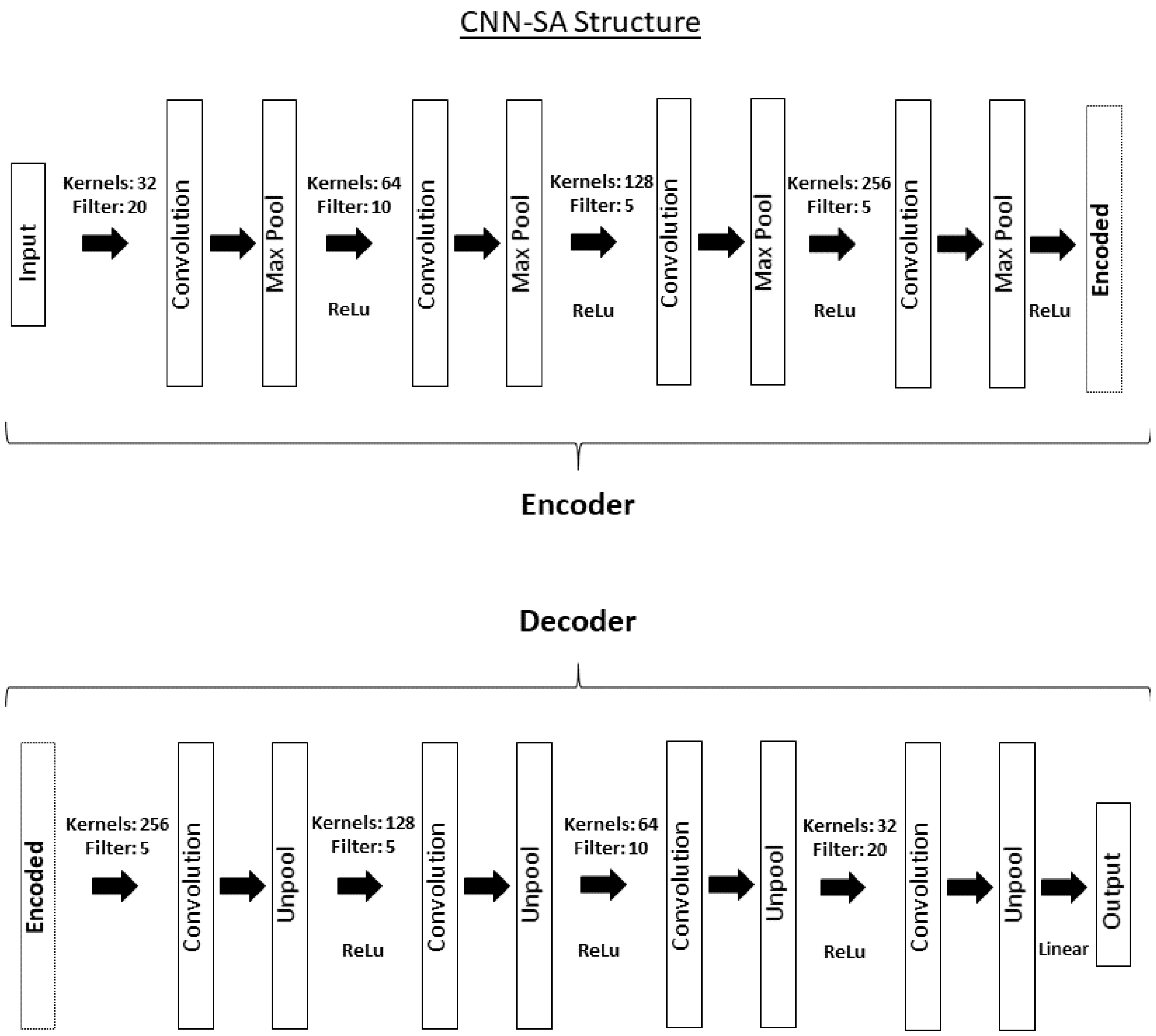


Figure 3.1: Schematic representation of the CNN-SA model architecture. The first eight-layers of the CNN-SA represent the encoder that creates a sparse representation of the input data. The last eight-layers of the model represent the decoder that takes the encoded data as input and then decodes it back into the original input dimensionality.

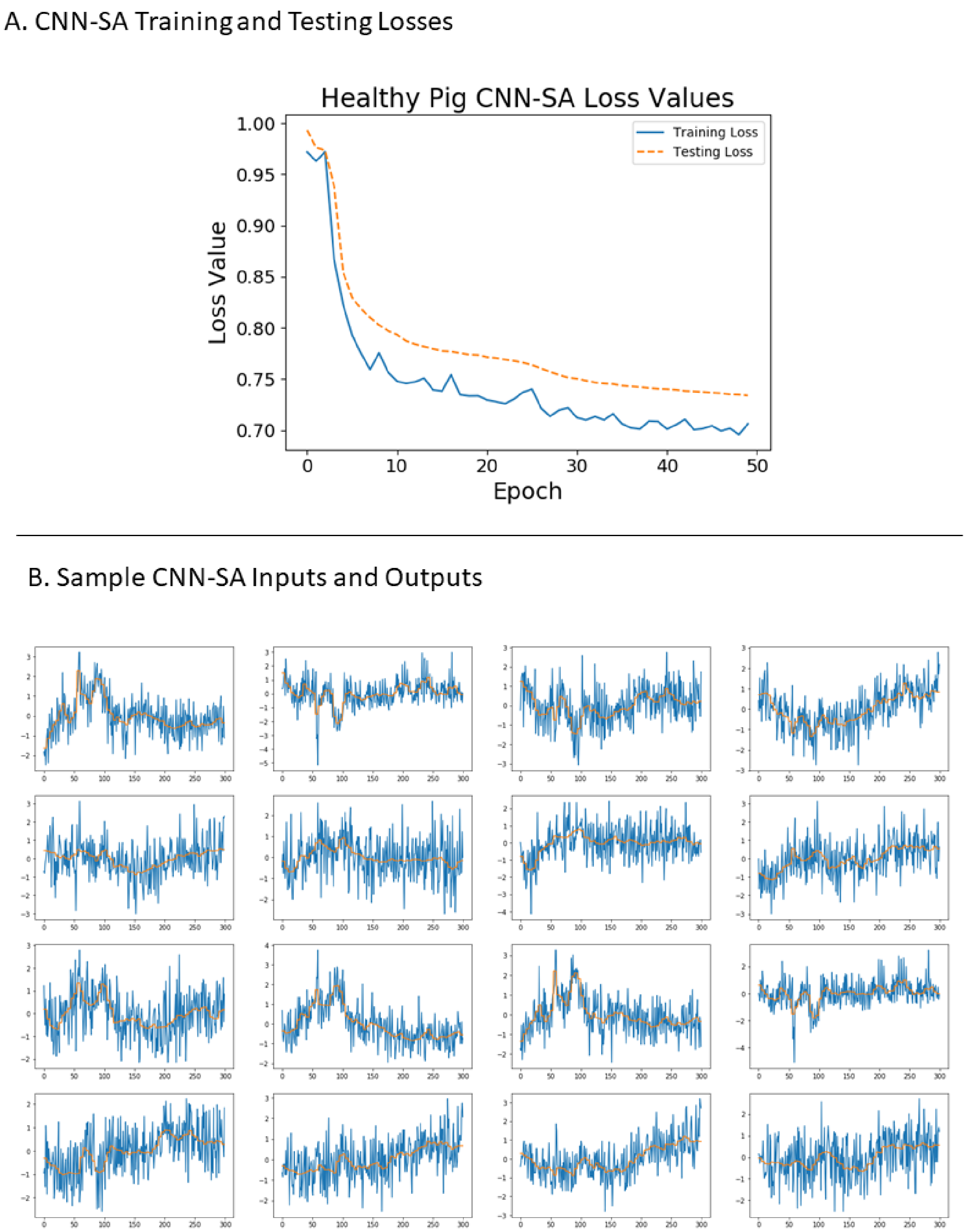


Figure 3.2: A.) Plot of the training (blue) and validation (gold) loss values over 50 epochs. B.) Examples of input time-series (blue) with decoded outputs of the CNN-SA overlaid (gold).

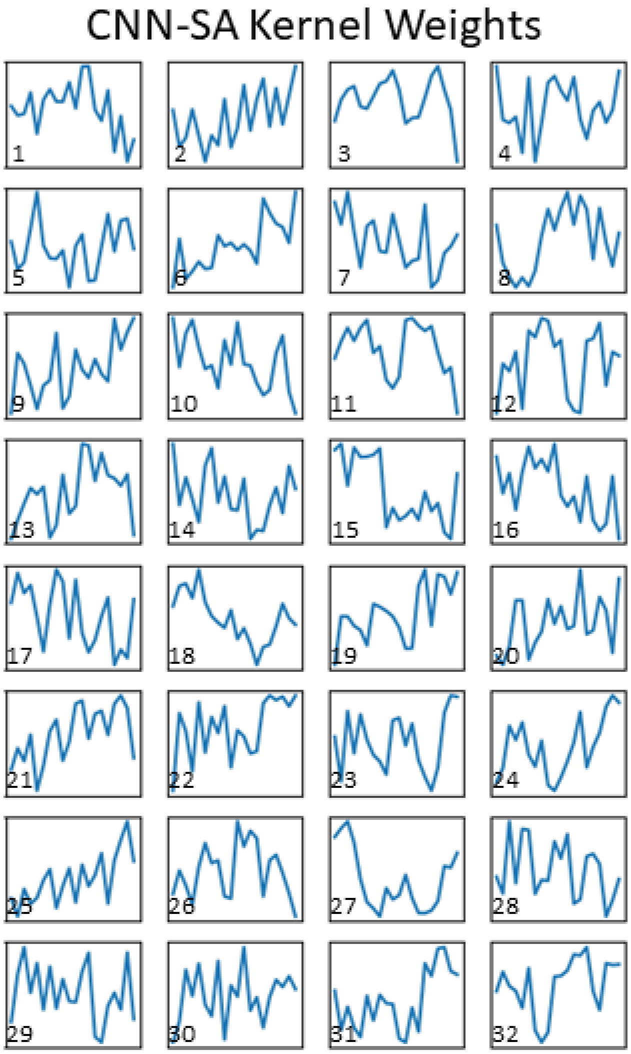
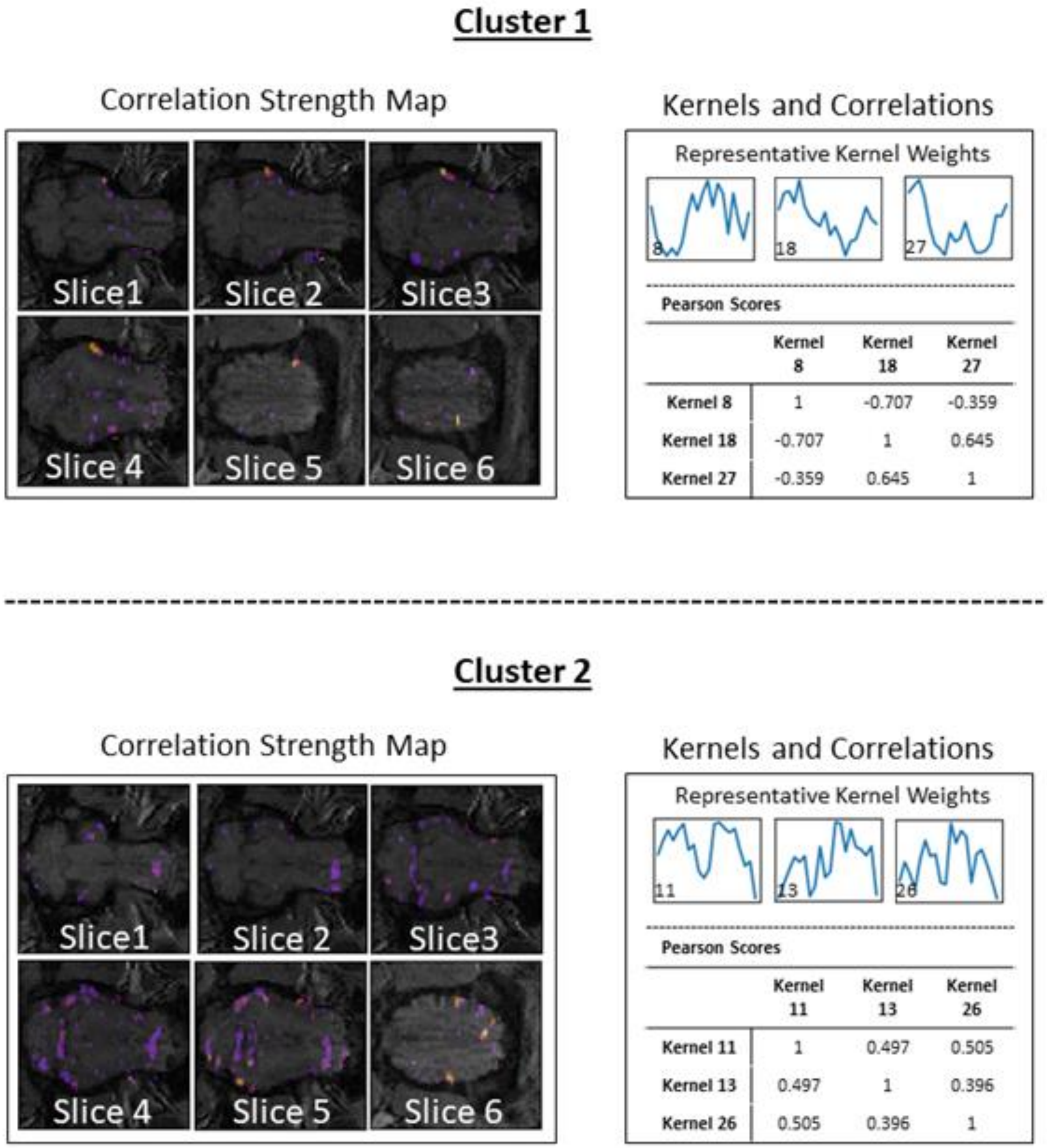


Figure 3.3: Plots of the 32 kernel weights from the first layer of convolution in the CNN-SA. Each kernel is sequentially labeled 1-32 for identification.



| | Kernel 8 | Kernel 18 | Kernel 27 |
|---|---|---|---|
| Kernel 8 | 1 | -0.707 | -0.359 |
| Kernel 18 | -0.707 | 1 | 0.645 |
| Kernel 27 | -0.359 | 0.645 | 1 |

| | Kernel 11 | Kernel 13 | Kernel 26 |
|---|---|---|---|
| Kernel 11 | 1 | 0.497 | 0.505 |
| Kernel 13 | 0.497 | 1 | 0.396 |
| Kernel 26 | 0.505 | 0.396 | 1 |

Figure 3.4: Correlation strength maps overlaid upon the pig anatomy for both cluster types found by analyzing kernel weights. Six of the most interesting slices that typify the individual cluster types are shown. Beside each of the clusters are the representative kernel weights that produce these clusters as well as Pearson coefficient scores between each of these kernel weights.

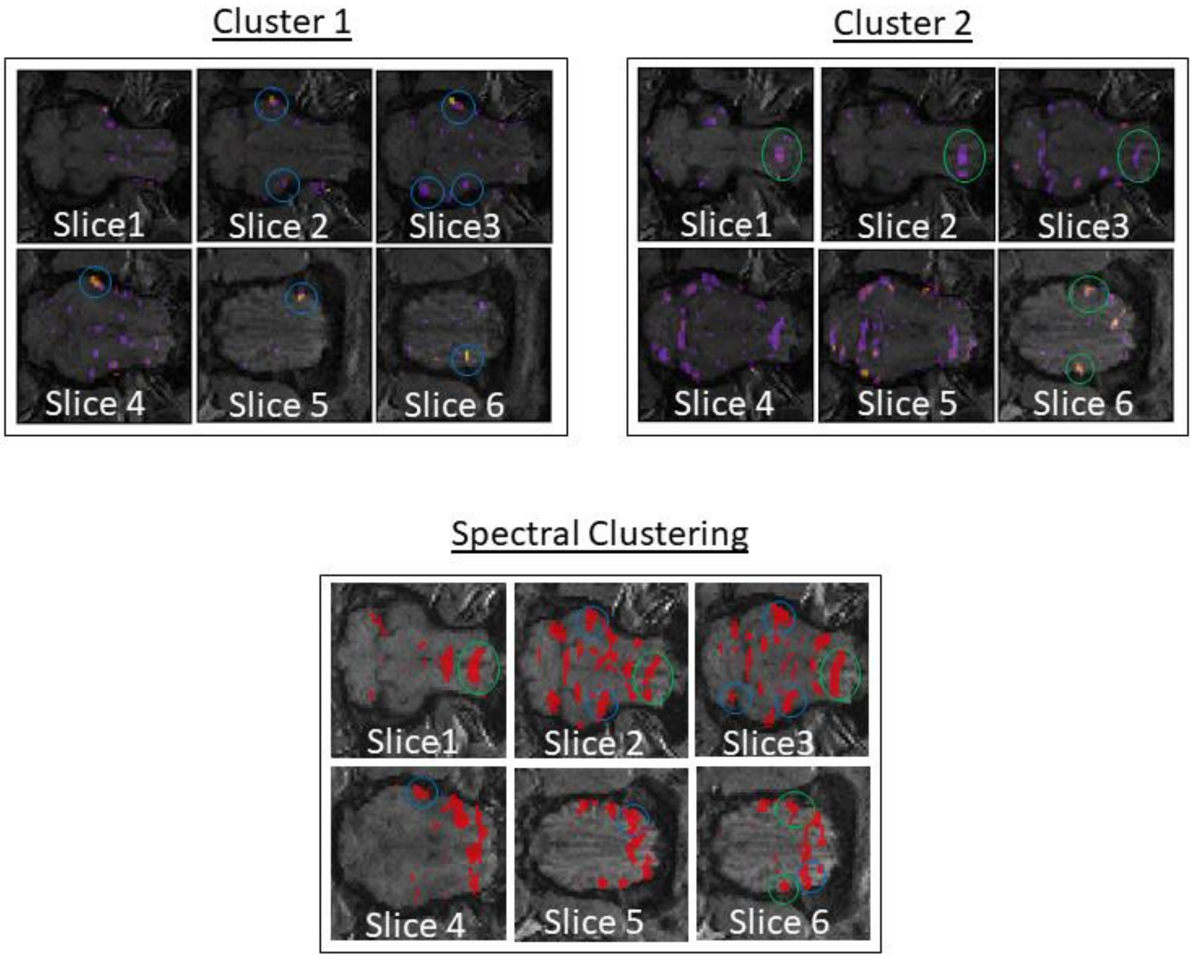


Figure 3.5: Top two plots display the original *Cluster* 1 and *Cluster 2* from the CNN-SA. The bottom plot shows spectral clustering results covering six similar slices as shown in the correlation strength maps. The red color simply shows locations of clusters and does not signify intensity. *Cluster 1* shows locations circled in blue that match with blue circled locations in the spectral clustering plot. *Cluster 2* shows locations circled in green that match with green circled locations in the spectral clustering plot.

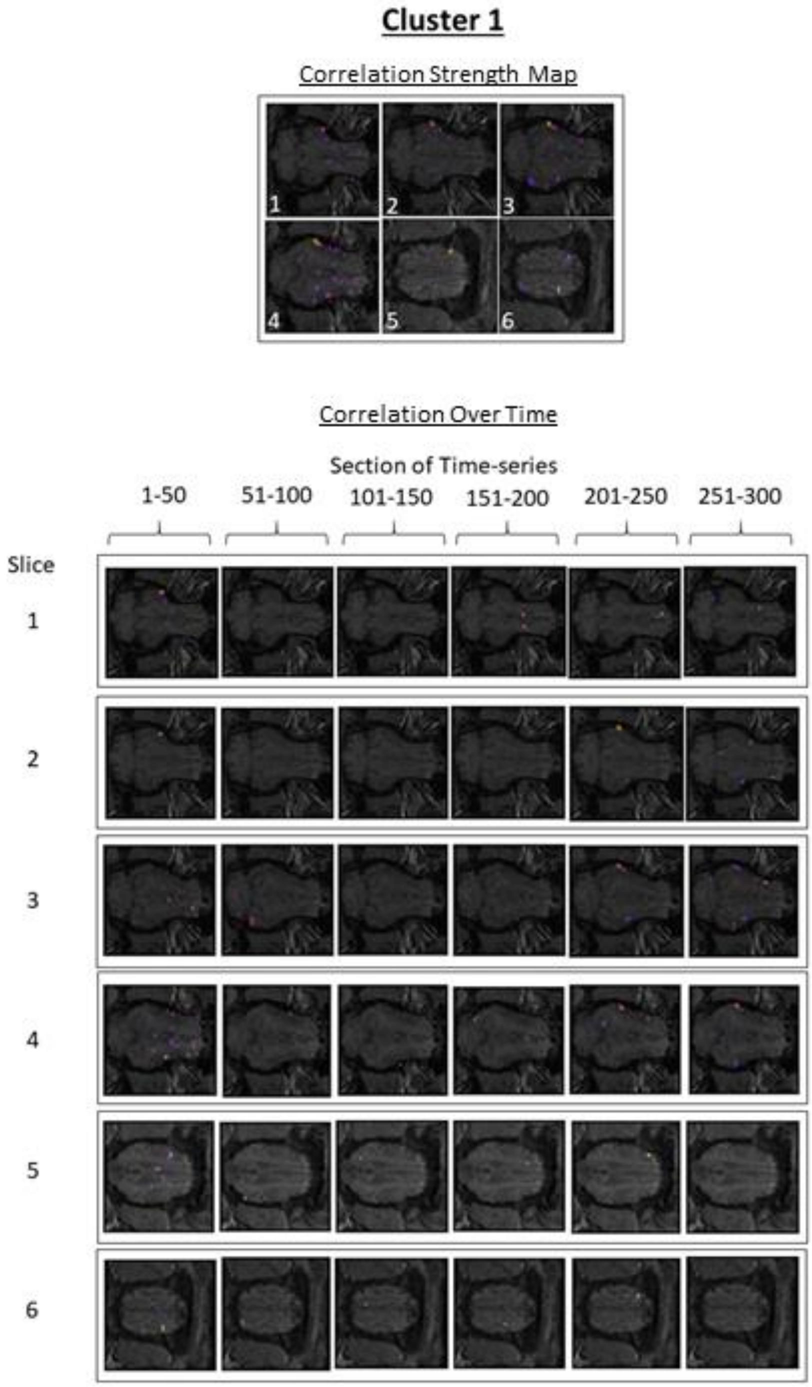


Figure 3.6: The correlation strength map for *Cluster 1* is displayed with each slice labeled. Below temporal correlation strength maps for the six subsections of time are displayed. Each row displays a unique slice while columns are subsections of time.

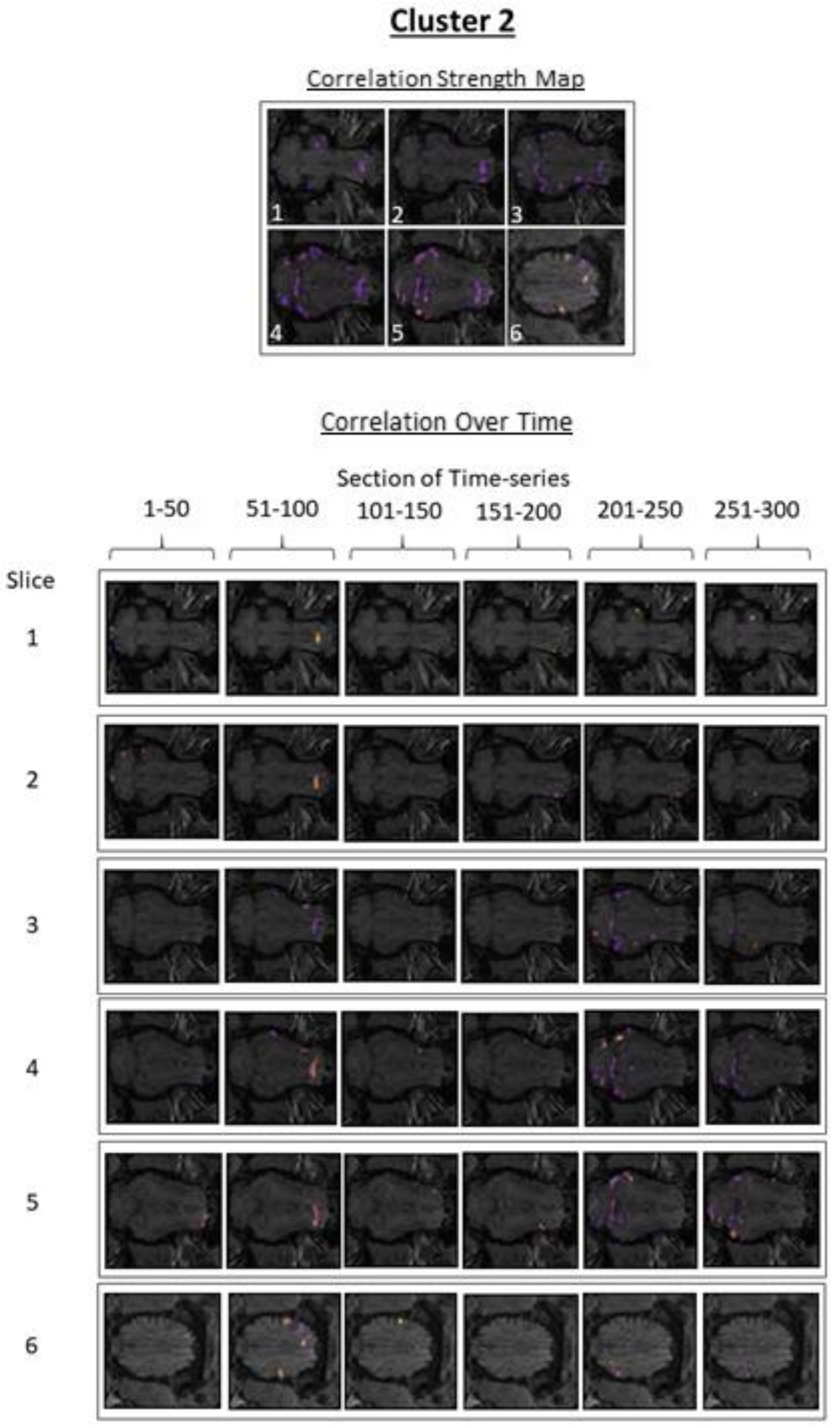


Figure 3.7: The correlation strength map for *Cluster 2* is displayed with each slice labeled. Below temporal correlation strength maps for the six subsections of time are displayed. Each row displays a unique slice while columns are subsections of time.

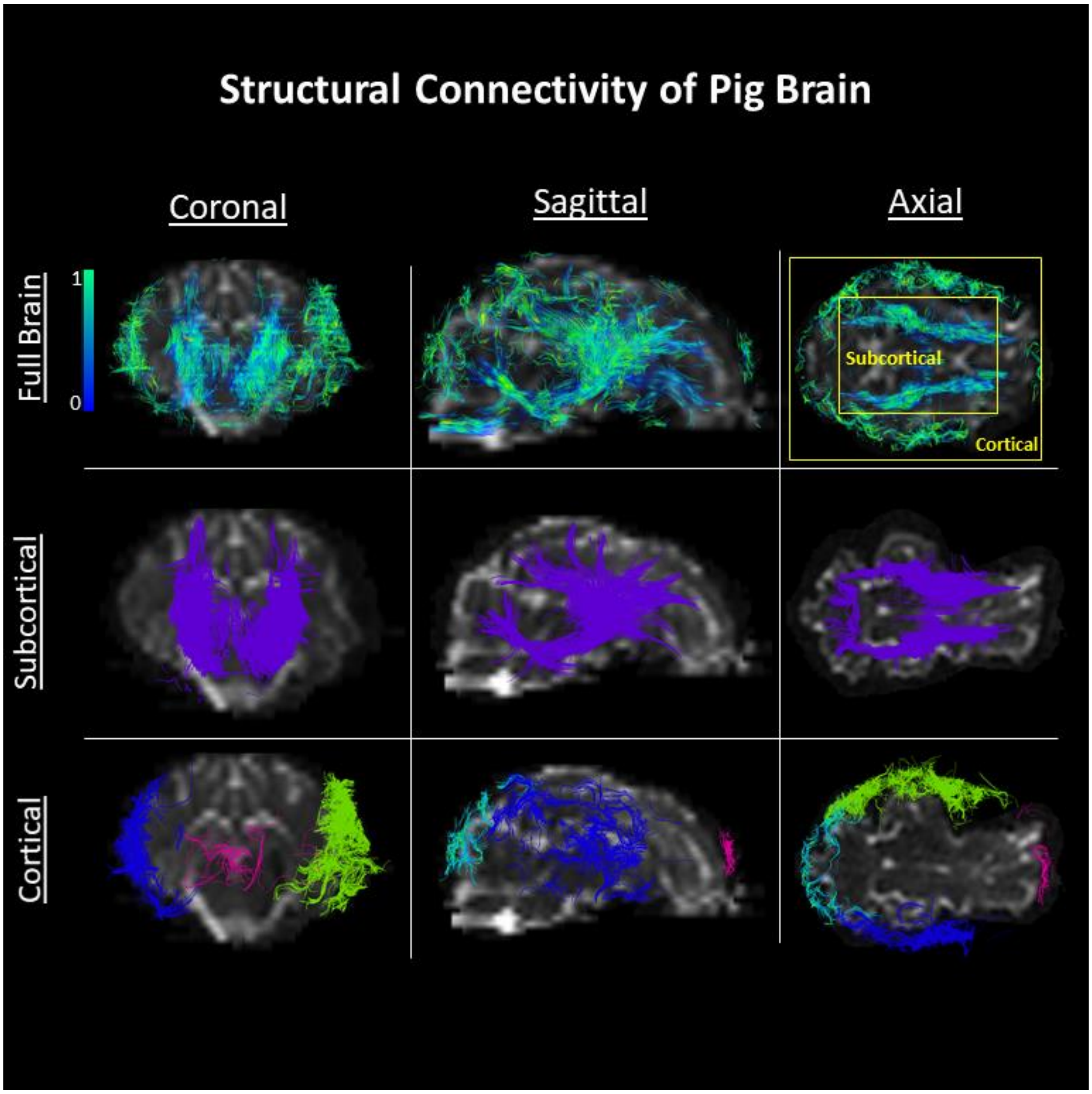


Figure 3.8: Plots of tractography fibers provided from the MedInria software. Columns correspond to three different views of the brain: coronal, sagittal and axial views. The first row shows all the fibers presented throughout the pig brain. The colors present in the fibers are results from FA calculations, where yellow corresponds to values closer to 1 while green corresponds to values closer to 0. The second two rows show only the subcortical or cortical fiber connections. The colors present in these figures have no meaning except to separate the fibers from the background of the anatomy.

# CHAPTER 4

## Conclusion

Deep learning has provided many different research opportunities in a wide range of scientific and commercial fields, and MRI provides many examples for implementing these machine learning techniques. The studies presented in this thesis have shown both the accuracy and versatility that deep learning can provide. Additionally, they have made significant advancements toward improving the analysis of MRI signal data for classification and brain functional connectivity analysis.

For classification tasks (Chapter 2) it was shown that CNNs can reach high accuracy values in adipose tissue differentiation using datasets commonly used for this task. More importantly it was shown that the same model architecture could provide the same level of accuracy on unprocessed data directly from the MRI scanner. This result opens new possibilities for rapid volume estimates of adipose tissue in the body.

The versatility of CNNs was shown in the creation of a new method for unsupervised brain functional connectivity analysis (Chapter 3) in pigs. While further confirmation of the accuracies of this method are needed, current evidence shows that this method can provide comparable results to spectral clustering while adding further modes of analysis. The ability to understand not only the spatial locations of functional connectivity but also the temporal organization of these connections will be very important for the future understanding how the pig brain functions.