Arab American University of Palestine
Faculty of Graduate Studies

A Convolutional Neural Network Framework for the Identification of Individual Neuronal Biomarkers based on Functional Brain Connectivity using Magnetoencephalography

By
Farah Abdellatif
Supervisor
Dr. Ahmad Hasasneh
Co- Supervisor
Dr. Jürgen Dammers, Research Centre Jülich, Germany

This thesis was submitted in partial fulfillment of the requirements for the Master`s degree in Data Science and Business Analytics

December/ 2021

A Convolutional Neural Network Framework for the Identification of Individual Neuronal Biomarkers based on Functional Brain Connectivity using Magnetoencephalography

By

Farah Abdellatif

This thesis was defended successfully on September 27th and approved by:

Committee members

Name	Signature
Dr. Majdi Owda	majdi Owda …………………………………..
Prof Labib Arafeh	…………………………………..
Dr. Ahmad Hasasneh	…………………………………..
Dr. Jürgen Dammers	…………………………………..

A Convolutional Neural Network Framework for the Identification of Individual Neuronal Biomarkers based on Functional Brain Connectivity using Magnetoencephalography
Farah Mohammad Waseem Abdellatif
2021

Acknowledgments

Data was provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

Thanks to Praveen Sripad (Forschungszentrum Jülich – INM4) for his valuable contribution in preprocessing the magnetoencephalography data.

Dedication

I dedicate this work to my beloved family and friends. A special feeling of gratitude goes to my parents, Mohammad and Nadira Abdellatif, who instilled the meaning of excellence and creativity in my heart.

I also dedicate this work to my respectful supervisors under whose constant guidance and support I was only able to complete the thesis.

Abstract

Finding the neuronal biomarkers at the individual level is an overarching objective in neuroscience research. Neuroscientists were able to identify individuals based on their brain functional and structural connectivity as depicted from their magnetoencephalography (MEG), functional Magnetic Resonance Imaging (fMRI), or electroencephalogram (EEG) scans. This individual identification process is also referred to in research as *individual brain fingerprinting*. In this study, resting state MEG (rMEG) data of healthy individuals is provided by the Human Connectome Project (HCP). A novel approach is introduced towards individual brain fingerprinting by applying a deep similarity learning model, a so-called *Siamese neural network* including convolutional neural networks (CNNs), to functional brain connectivity (FC) metrics from rMEG. To prove the superiority of the deep learning approach, the performance is compared against a sophisticated machine learning algorithm, Support Vector Machine (SVM). Coherence and Amplitude Envelope Correlation (AEC) were used as the FC metrics for the SVM and the Siamese network. The Siamese network was able to outperform the machine learning model with an accuracy of 97% as compared to 81% accuracy coming from the SVM. In conclusion, convolutional neural networks are a very powerful computational tool that can boost analysis of any field, such as neuroscience. To this date, the research pool for FC fingerprinting has not yet benefited from the deep learning techniques, where this research can be used as the first step towards enhancing the analysis of FC fingerprinting.

Keywords: Deep Learning, Magnetoencephalography, Siamese Neural Network, Functional Connectome Fingerprinting, Convolutional Neural Network.

A Convolutional Neural Network Framework for the Identification of Individual Neuronal
Biomarkers based on Functional Brain Connectivity using Magnetoencephalography
Farah Mohammad Waseem Abdellatif
2021

Table of Contents

THESIS OVERVIEW 10

1.1 INTRODUCTION 10

1.2 PROBLEM STATEMENT 11

1.3 RESEARCH OBJECTIVES 11

1.4 THESIS ORGANIZATION 11

BACKGROUND AND RELATED WORK 13

2.1 MEG FUNCTIONAL CONNECTOME FINGERPRINTING 13

2.1.1 Theoretical Background 13

2.1.2 Literature Review 14

2.2 DEEP LEARNING FOR FC FINGERPRINTING 16

2.2.1 Theoretical Background: Convolutional Neural Networks 16

2.2.2 Theoretical Background: Siamese Neural Network 28

2.2.3 Theoretical Background: Support Vector Machine 30

2.3 LITERATURE REVIEW 32

2.4 SUMMARY 35

DEEP LEARNING MODEL DESIGN 37

3.1 INTRODUCTION 37

3.2 ABOUT THE DATASET 37

3.3 DATA PROCESSING 37

3.4 THE SIAMESE NEURAL NETWORK 40

3.4.1 Training the Neural Network 40
3.4.2 Most Promising Architecture 41
3.5 SUMMARY 42
RESULTS AND DISCUSSION 44
4.1 FEATURE EXTRACTION 44
4.2 SIAMESE NEURAL NETWORK MODEL RESULTS 46
4.3 SUPPORT VECTOR MACHINE RESULTS 49
4.4 SUMMARY 50
CONCLUSIONS AND FUTURE WORK 51
5.1 CONCLUSION 51
5.2 CONTRIBUTION AND POSSIBLE APPLICATIONS 51
5.3 FUTURE WORK 51
5.4 CLOSING REMARKS 52
BIBLIOGRAPHY 53
ABBREVIATIONS 62
ملخص الرسالة 64

List of Tables

TABLE 1: NOTATION FOR FORWARD PROPAGATION 23

TABLE 2: PSEUDO CODE SAMPLE FOR TRAINING A NEURAL NETWORK 26

TABLE 3: SIAMESE NETWORK RESULTS SHOWING PREDICTION ACCURACY AND LOSS FOR NORMALIZED AND UNNORMALIZED DATA 47

TABLE 4: SVM RESULTS SHOWING PRECISION, RECALL, F1-SCORE, AND ACCURACY FOR NORMALIZED AND UNNORMALIZED DATA 49

TABLE 5. TERMINOLOGY AND DEFINITIONS 62

List of Figures

Figure 1: Architecture of a Convolutional Neural Network including convolution, pooling, and fully connected layers. Adapted from (Alom et al., 2019). 17

Figure 2: Example of a convolutional layer of size (5x5) and a kernel of size (3x3) 18

Figure 3: Example of a convolutional layer with padding and stride 19

Figure 4: Example of average and max pooling layers with input size (4x4) and pooling window size (2x2) 20

Figure 5: Example for data transformation by ReLu and Sigmoid activation functions. Adapted from (Mash et al. ,2016) 21

Figure 6: A simple representation of a 2-layer neural network showing differences between forward and backward propagation 25

Figure 7: A simple representation of a Siamese Neural Network 30

Figure 8: Data Processing Block Diagram. The rMEG scans from the HCP are preprocessed to compute the coherence and AEC. The FC data is then further processed to be used in proper format as input for subsequent machine and deep learning analysis 38

Figure 9: Flowchart for training the Siamese Neural Network 40

Figure 10: FC image pairs are fed into the twin CNNs. Each CNN has 2 hidden layers of convolution followed by Averaged pooling and then two fully

connected layers. Outputs from the CNNs are merged via the Euclidean distance and then contrastive loss gives the similarity result 42

Figure 11: Comparison of normalized and unnormalized FC data 45

Figure 12: Min-max normalized FC values over the 7 frequency bands for a single subject 46

Figure 13: Siamese Network evaluation metrics for data normalized and unnormalized data 48

THESIS OVERVIEW

1.1 Introduction

Data science involves the extraction, organization, and analysis of large data sets which can be applied to a huge variety of fields including neuroscience. In fact, machine learning and deep learning techniques have boosted the analysis in the field of neuroimaging. For example, Multivariate Pattern Analysis (MVPA) is used in functional Magnetic Resonance Imaging (fMRI) data to solve problems such as classification of facial expressions (Liang et al., 2018) and classification of primary insomnia (C. Li et al., 2019). Additionally, the deep learning framework has proved its superiority to classical machine learning for processing large amounts of spatiotemporal neuroimage data which allows for solving complex problems such as neurological diagnosis and classification (Gerven & Bohte, 2018).

Neuroscientists have long recognized that individuals of the same neurologically healthy group show variability in the brain function and structure. The variations in the brain function have been associated with the individual’s traits and behavior, resulting in neuronal fingerprints that are substantial and reproducible for serving as individual biomarkers (Da et al., 2021). Neuroscience research has shown great promise in inferring the individual neural fingerprint from the functional connectivity derived from the magnetoencephalography (MEG), Electroencephalogram (EEG), and fMRI data (Amico & Goñi, 2018; Barcha et al., 2014; Da et al., 2021; Kong et al., 2019; Sareen et al., 2021). While existing research presented well-performing individual identification classification models, the methods used were limited to classical statistical and machine learning algorithms. To our knowledge, a deep learning approach has not yet been investigated for the domain of identifying the individual fingerprint based on the brain functional connectivity. The research quest of applying a sophisticated deep learning algorithm towards enhancing the accuracy and robustness of these models remains open for us to explore in this thesis.

1.2 Problem Statement

Individuals of the same neurologically healthy group are diverse in terms of personality traits and behavior. Many studies have provided evidence that this individual variability is reflected in the statistical variability of the individual’s neural data (Amico & Goñi, 2018; Barcha et al., 2014; Kong et al., 2019). For this research, the individuals shall be identified based on the features from their brain functional connectivity derived from resting-state MEG (rMEG) as in (Da et al., 2021). However, our proposed identification technique will make use of a novel approach, which is a deep learning model, such as a convolutional neural network (CNN).

1.3 Research Objectives

This research aims to introduce a novel approach towards identifying individuals based on their neuronal biomarkers, so a deep similarity learning model known as a Siamese neural network (Bromley et al., 1993) is used. It also aims to prove the superiority and robustness of deep learning over other machine learning techniques for this domain by comparing the performance of the Siamese network to SVM algorithm.

1.4 Thesis Organization

The following chapter includes background and literature review of related work, where the content first introduces the reader into the neuroscientific background and then transitions into justifying the use of the deep learning approach. Later in Chapter 3, the design of the deep learning model is presented starting from data acquisition to the neural network. Results are presented and discussed in Chapter 4, and the thesis is concluded in Chapter 5.

BACKGROUND AND RELATED WORK

2.1 MEG Functional Connectome Fingerprinting

2.1.1 Theoretical Background

MEG is considered as a patient-friendly neuroimaging method because of its non-invasive nature (Hamalainen et al., 1993). It is a biomedical functional brain scanning technique that uses sensitive magnetometers to record magnetic fields generated by neural activity. Since the magnetic fields are not distorted by scalp or skull, MEG has the advantage over other types of techniques in providing a highly accurate temporal and spatial resolution of the neuronal activity (Filippi, 2015; Singh, 2014).

After MEG data acquisition, researchers can choose from several field-related data analysis steps to apply. To study the communication within and across the brain, the brain networks are quantified according to three types: Structural, Functional and Effective (Sakkalis, 2011). Where, structural connectivity analysis is used to detect the anatomical brain network maps that connect the brain regions. Functional Connectivity (FC) analysis is used to identify which brain regions are connected, i.e., sharing information at different frequency bands. Lastly, Effective Connectivity (EC) analysis is used to determine the direction of the information flow between brain regions (Bowyer, 2016).

Over the past few decades, many FC measures have been proposed and evaluated for electrophysiological data. FC measurements can be analyzed in the frequency domain or in the time domain (Brookes et al., 2011; Gross et al., 2013). The well-known coherence metric is used as a frequency domain measure, and Amplitude Envelope Correlation (AEC) is used as a time domain measure, where:

- ***Coherence*** is considered the most common measure to investigate the connectivity between brain regions for decades (Papanicolaou et al., 2020). It estimates the degree of linear association between the phases and amplitudes of two time series signals within nonoverlapping frequency bands (f) (Koopmans, 1995):

$$C_{xy}(f) = \frac{\left|S_{xy}(f)\right|^2}{\left|S_{xx}(f)\right| \cdot \left|S_{yy}(f)\right|} \qquad (1)$$

, where:

    - S_{xy} is cross spectrum density and S_{xx}, S_{yy} are the autospectral densities of the two time series signals x and y.
    - The result is a symmetric matrix with values $0 \leq C_{xy} \leq 1$, where a value of 1 indicates perfect coupling and 0 represents no relationship.

- ***AEC*** is a measure of the correlations between the Amplitude Envelopes (AE, i.e., amplitude fluctuations in an oscillation over time) of two oscillatory brain signals, where, AEC also ranges from 0 to 1, which means that high AEC values indicate synchronous AE fluctuations (Zamm et al., 2018).

2.1.2 Literature Review

Neuroscience research has shown great promise in inferring the individual neural fingerprint from the Functional Connectome profiles derived from the neuroimages. The methods to extract the identity have varied according to the type of data used.

For example, MEG-based features were used in (Da et al., 2021; Demuru et al., 2017; Sareen et al., 2021). In (Demuru et al., 2017), researchers investigated the resemblance between twins from monozygotic (MZ) pairs using rMEG by examining similarities in the functional

(FC) or effective (EC) connectivity patterns. The connectivity measures used included AEC amongst others. The measure for similarity relied on the Spearman’s rank correlation coefficient.

As for (Da et al., 2021), researchers performed individual identification across pairs of MEG data segments using either the functional or spectral connectivity features. The measure used for FC was AEC, and Power Spectral Density (PSD) was used for spectral fingerprinting. Identifying an individual was possible by finding the highest value in the Pearson’s correlation coefficient matrix for the feature (FC or PSD), where the corresponding column index determined the predicted identity.

On the other hand, researchers of (Sareen et al., 2021) ran a deeper investigation into what influences the FC fingerprint. They studied the impact of the frequency bands (Delta, Theta, Alpha, Beta, and Gamma) on FC identification. They also investigated which brain connections contribute more to the FC identifiability. Lastly, they compared the FC fingerprinting feature patterns from the MEG data to those obtained from fMRI images of the same subject. The identifiability matrix for each of the FC measures was constructed based on Pearson’s correlation coefficient.

Another type of neuroimaging that is vastly used in the research domain of FC fingerprinting is resting-state and behavioral fMRI, such as (Amico & Goñi, 2018; Finn et al., 2015). In (Finn et al., 2015), the FC of the target subject was identified by computing the Pearson correlation coefficient of the target FC against a sample database of FCs. As for (Amico & Goñi, 2018), researchers aimed to improve the self-identifiability results of the previous work of (Finn et al., 2015). This was possible by first reconstructing the individual connectivity profile using Principal Component Analysis (PCA) to extract different components associated with the whole population, and then, the Pearson correlation coefficient for the FC was computed to identify the individual.

The neuronal fingerprinting models of (Amico & Goñi, 2018; Da et al., 2021; Demuru et al., 2017; Finn et al., 2015) showed very high and acceptable accuracy. Also, results were consistent in showing that that there is no association between the duration of the scan and the identifiability. Additionally, researchers verified that several variables did not contribute to self-identifiability, namely, motion, cardiac and ocular artifacts, sample heterogeneity (i.e., restricting fingerprinting to healthy participants), and the individual's demographics such as age, sex, and handedness.

With regards to (Sareen et al., 2021), their results showed that there is a high variability of functional connectivity measures across the tested frequency bands, where the Alpha and Beta frequency bands showed the best results.

While the results of all cited papers above have shown high accuracy and robustness, they have relied solely on conventional statistical and machine learning methods. Therefore, this creates a research opportunity to explore the performance, sensitivity, and specificity of deep learning methods for the identification of individuals neuronal biomarkers.

2.2 Deep Learning for FC Fingerprinting

2.2.1 Theoretical Background: Convolutional Neural Networks

Pattern recognition tasks, such as classification, fall under the category of supervised learning since they require labeled data for all classes in order to derive a discrimination function (Japkowicz, 2001). Artificial Neural Networks (ANN) are deep neural networks with many layers able to handle huge amount of data to perform supervised deep learning such as regression analysis, and classification tasks ("Artificial Neural Networks," 2021). One of the most popular ANNs are the ***Convolutional Neural Networks*** (CNNs) (LeCun, 1989) that have been extremely successful for pattern recognition applications (Basu et al., 2010). CNNs impressive ability to process images for pattern recognition allowed for it to be a winning design to classify the largest image dataset of the 2012 ImageNet challenge (Krizhevsky et al., 2012).

Structure of the Convolutional Neural Network

Generally speaking, a CNN is a deep neural network with multiple layers, namely, convolution, pooling, non-linearity, and fully connected layers (Alom et al., 2019). ***Figure 1*** shows a representation for a multi-layered CNN with convolution and pooling layers for feature extraction, and fully connected layers for the classification. Each layer is defined below in greater details.

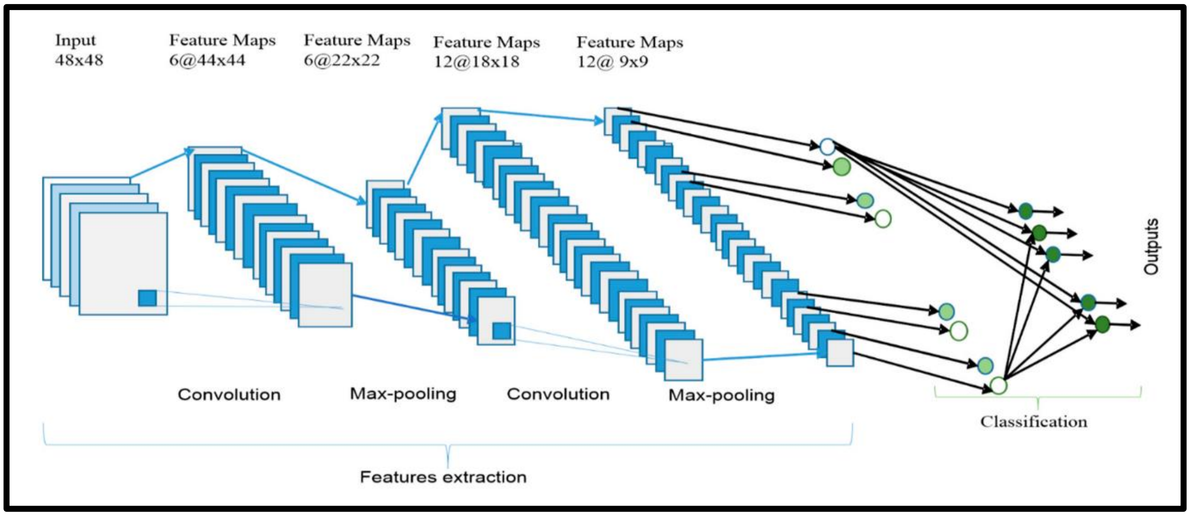


Figure 1: Architecture of a Convolutional Neural Network including convolution, pooling, and fully connected layers. Adapted from (Alom et al., 2019).

The ***Convolution*** layer is the core building block of a CNN. Convolution is a mathematical operation where a matrix of weights called a kernel traverses the image to transform it. The output which is referred to as the feature map is the dot product of the filter weights of the kernel and the values of the input data matrix (Albawi et al., 2017). Refer to ***Figure 2*** for an illustration of the convolution process.

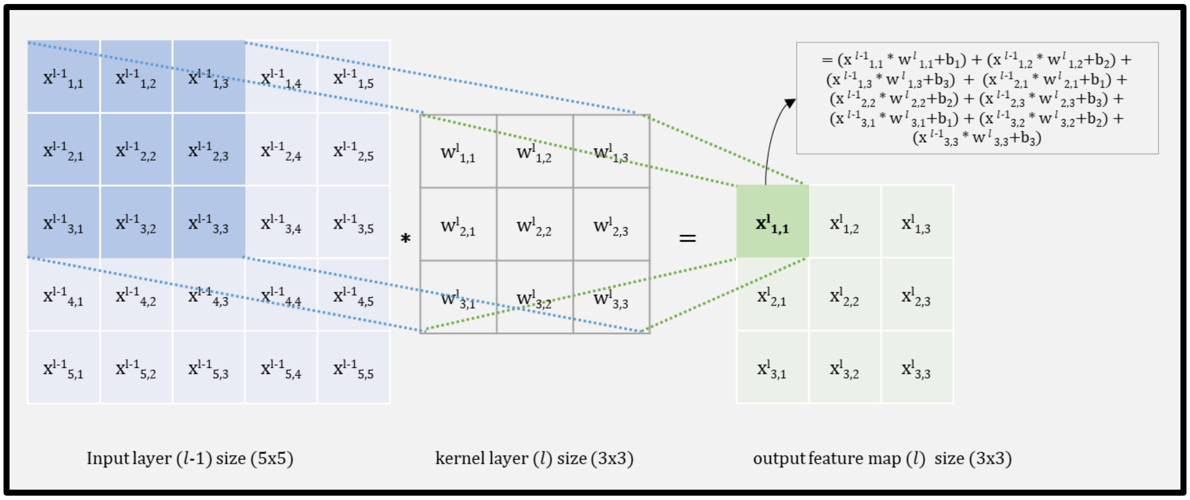


Figure 2: Example of a convolutional layer of size (5x5) and a kernel of size (3x3)

The convolutional process is represented by equation (2) (Alom et al., 2019), where:

- x_j^l is the output of layer l, and x_i^{l-1} is the output of layer $l-1$
- w_{ij}^l and b_j^l are the weights and biases which are defined by the CNN
- M_j represents the input layer map

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{(l-1)} * w_{ij}^l + b_j^l\right) \qquad (2)$$

Padding and stride are two other parameters that impact the size and shape of the feature map. Padding is used when it is desired to have a convolution output of the same size as the input, so zeros are padded to the input. Stride is the number of rows and columns in the input image passed over by each slide of the kernel horizontally and vertically (Nielsen, 2015). Refer to ***Figure 3*** for an illustration of padding and stride. The size of the output feature map can be calculated according to equation (3), and the number of rows and columns needed for padding, can be calculated according to equation (4) (Alom et al., 2019), where:

- m is the dimension of the output feature map?
- n is the dimension of the input
- f is the dimension of the kernel
- s is the stride value
- p is the number of rows and columns needed for padding

.

$$m = \frac{n - f}{s} + 1 \tag{3}$$

$$p = \frac{f - 1}{2} \tag{4}$$

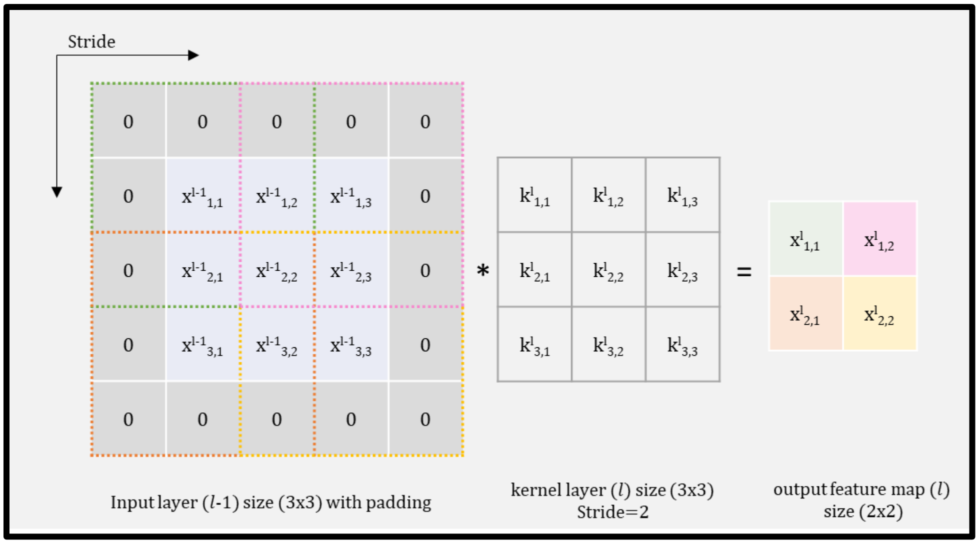


Figure 3: Example of a convolutional layer with padding and stride

The ***pooling layer*** is a subsampling layer where a pooling window traverses the input. The pooling layer output is deterministic depending on the type of the pooling layer; it computes the maximum or the average value of the pooling window (Albawi et al., 2017). The pooling function is represented by equation (5), where the down(.) function represents the subsampling process (Alom et al., 2019):

$$x_j^l = down\left(x_j^{l-1}\right) \tag{5}$$

For example, if the input has size $m \times m$, and the pooling window has size $n \times n$ then the pooling layer reduces the input layer size by a factor of N: $\left(\left(\frac{m}{n}\right) \times \left(\frac{m}{n}\right)\right)$. Refer to **Figure 4** for an illustration of the pooling layer.

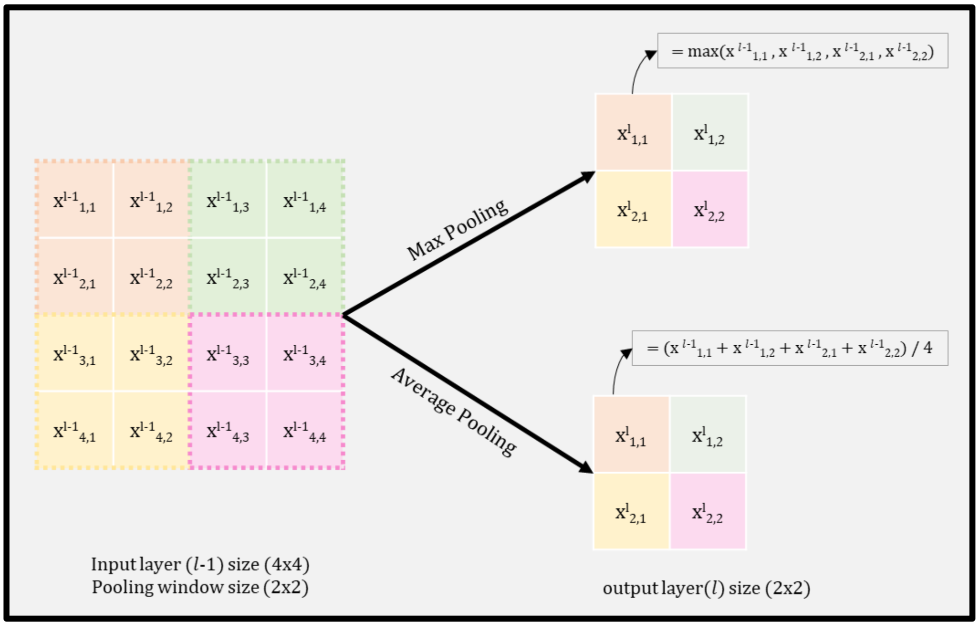


Figure 4: Example of average and max pooling layers with input size (4x4) and pooling window size (2x2)

Looking at the convolutional and pooling layers from a wider perspective, the CNN capability to convolve can be interpreted as sliding filter that extracts local features from the input data. In addition, hierarchical stacking of the convolutional layers combined with pooling layers allows for extracting global features from the input (Bock et al., 2021).

Furthermore, the convolutional and pooling layers can be considered as linear systems since their outputs are geometric transformations of the inputs. These layers lack the ability to express the non-linearities existing in the data (Zoumpourlis et al., 2017). Hence, the non-linearity layer is a crucial part of the CNN.

The ***non-linearity layer,*** also referred to as a ***non-linear activation function***, is used to apply a nonlinear transformation to data which allows the CNN to learn and recognize complex mappings from data (Mash et al., 2016). For example, the convolutional layer produces the dot product of sub-samples from the input data and the kernel, and then an activation function is used to get a transformed nonlinear output of the convolutional layer and supply it as input to the next layer(Sharma et al., 2020). It is important to note that the activation function does not change the size of the data. Two activation functions are are displayed in **Figure 5**: Rectified Linear Unit (ReLu) and Sigmoid which.

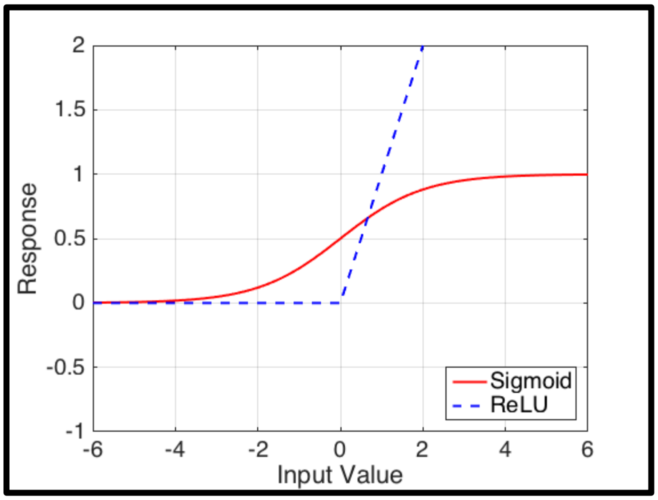


Figure 5: Example for data transformation by ReLu and Sigmoid activation functions. Adapted from (Mash et al. ,2016)

ReLu layer for non-linearities has been highly successful for computer vision tasks and proved to be faster in training the CNN as compared to other activation functions (Dahl et al., 2013). ReLu activation function is a simple calculation that returns the value provided as input directly, or a value 0 if the input is 0 or less. Equation (6) notes the formula to transform the data with ReLu function (Mash et al., 2016):

$$y = max\{0, x\} \tag{6}$$

The Sigmoid activation function, also called the logistic sigmoid function, maps data into a small range between 0 and 1, which useful for interpreting output as a probability (Pratiwi et al., 2020). Equation (7) notes the formula to transform the data with Sigmoid function (Mash et al., 2016):

$$y = \frac{1}{1 + e^{-x}} \tag{7}$$

Lastly, the ***fully connected layer*** refers to a layer where each neuron in it is directly connected to every neuron in both the previous and the next layers. This layer is useful at the end of the CNN model after the convolution, pooling and ReLu layers, to serve as the classification layer (Albawi et al., 2017).

Training the Convolutional Neural Network

In general, supervised feedforward neural networks involve an input vector and an associated response vector that represents the label. The common goal of the network is to use training data samples to make a function that approximates the label for unseen data during the training (Japkowicz, 2001). During the training process, the CNN fits its parameters, which are the weights and biases, to the data via learning algorithms. These learning algorithms compare a given label to the predicted label from the network function and then adapt the network parameters according to this comparison (Günther & Fritsch, 2010). A general flow for training the neural network includes the following steps (Günther & Fritsch, 2010; Karnin, 1990). A pseudo code summarizing this section is presented in **Table 2**:

1. Initial network weights and biases chosen randomly.
2. ***Forward propagation*** refers to the process of propagating the input vector from the input layer, through the hidden layers, to the output layer. During forward propagation, the intermediate outputs, weights and biases of the neural network are stored (J. Li et al., 2012). Based on equations (2,6,7), forward propagation for two hidden layers, as in **Figure 6**, can be represented in simplified notation as in **Table 1**.

Table 1: Notation for Forward Propagation

Description	Notation
Value of neuron at hidden layer 1	$z^2 = W^1x + b^1$
Activation function at hidden layer 1	$a^2 = f(z^2)$
Value of neuron at hidden layer 2	$z^3 = W^2a^2 + b^2$
Activation function at hidden layer 2	$a^3 = f(z^3)$
Output	$s = W^3a^3$

3. A ***cost function*** measures the error which is the difference between predicted and expected output. One of the most popular cost functions is the mean square error (MSE) (Wu, 2017). The cost function C can be denoted as a function of predicted output s and expected output y: $C = f(s, y)$
4. In case that the expected output is not achieved, a learning algorithm, such as back propagation is used. ***Back propagation*** is a method to monitor and enhance the

learning of the CNN. It relies on the *cost function* and the *gradient descent* to adjust the parameters of the network in order to achieve the minimum error (J. Li et al., 2012). The gradients for weights and biases are computed based on the chain rule in equations 8 and 9, and then the intermediate parameters are optimized as in equations 10 and 11, where η is the learning rate (J. Li et al., 2012):

$$\frac{\partial C}{\partial w_{ij}^{l}} = \frac{\partial C}{\partial z_i^l}\frac{\partial z_i^l}{\partial w_{ij}^{l}} = \frac{\partial C}{\partial z_i^l} a_j^{l-1} \tag{8}$$

$$\frac{\partial C}{\partial b_i^l} = \frac{\partial C}{\partial z_i^l}\frac{\partial z_i^l}{\partial b_i^l} = \frac{\partial C}{\partial z_i^l} \tag{9}$$

$$w \coloneqq w - \eta \frac{\partial C}{\partial w} \tag{10}$$

$$b \coloneqq b - \eta \frac{\partial C}{\partial b} \tag{11}$$

5. The process stops in case of ***convergence***, where all absolute partial derivatives of the cost function with respect to the weights and biases are smaller than a given threshold.

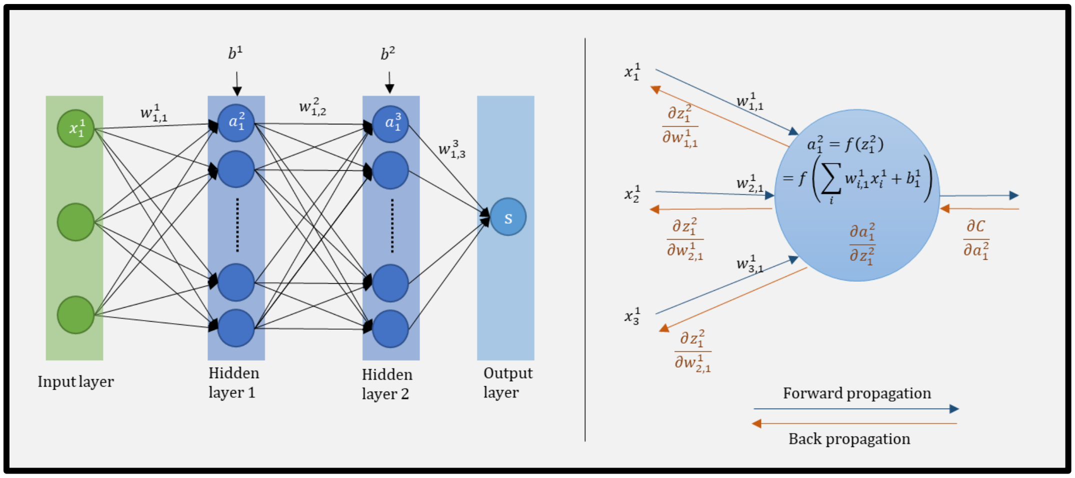


Figure 6: A simple representation of a 2-layer neural network showing differences between forward and backward propagation

Additionally, there are various techniques to improve the training of the deep learning model. These techniques include initialization, batch normalization, regularization, and optimization. Each method is described briefly below.

- The ***initialization*** of parameters has a big impact on the overall network accuracy. A popular initialization approach is the one proposed by (He et al., 2015). As expressed in equation 12, the distribution of the weights of the l^{th} layer is made to be normal (N) with a mean of zero and a unit variance of $\frac{2}{n_1}$, where n_l equals the number of input neurons of the l^{th} layer:

$$w_l \sim N\left(0, \frac{2}{n_l}\right) \tag{12}$$

- ***Batch Normalization*** is applied to internal layers of the network, where the features of a layer are independently normalized with a zero mean and a unit variance to eliminate the changes in the distributions of the internal layers. Batch normalization impacts the overall accuracy as it allows the network to converge faster and enhances regularization of the training (Ioffe & Szegedy, 2015).

- Dropout, proposed by (Hinton et al., 2012), is the simplest ***regularization method***, where a random subset of activations in a layer is set to zero. Dropout impacts the overall accuracy as it reduces overfitting.
- There are many ***optimization*** methods such as Adagrad, Adadelta, RMSprop, and Adam. The Adam optimizer was introduced by (Kingma & Ba, 2015) for calculating an adaptive learning rate based on the momentum and the magnitude of the gradient. As compared to previous optimization algorithms, Adam has proven to perform better for improving the overall accuracy and getting better convergence (Ruder, 2016).

Table 2: Pseudo code sample for training a neural network

Algorithm for training the neural network

```
//certain function should be predefined such as activation and loss functions
PROCEDURE Forward Propagation ()
   FOR each layer
    FOR each layer's neurons
      Set weighted Sum to 0 // initialization
      FOR each neuron's links
        Multiply link's weight with associated previous Layer's neuron's value and add bias
        Add result to weighted Sum
      END FOR
      Call activation function (weighted Sum)
```

Algorithm for training the neural network

```
    Set neuron's value to result
  END FOR
 END FOR
END PROCEDURE
//Error summation function during forward pass
PROCEDURE calculate Total Error ()
 Set total Error to 0
 Call loss function ()
 Add result to total Error
END PROCEDURE
```

```
//certain functions should be predefined such as the gradient chain rule
 PROCEDURE Back Propagation ()
 Get Total Error
 FOR each layer
   FOR each layer's neurons
     FOR each neuron's links
     Call Gradient Chain Rule function ()
     Set gradient to result
     Compute link's new weight as weight - (learning Rate * gradient)
     Compute link's new bias as bias - (learning Rate * gradient)
  END FOR
```

Algorithm for training the neural network

```
  END FOR
Overwrite weights and biases
END PROCEDURE
```

2.2.2 Theoretical Background: Siamese Neural Network

As mentioned previously, the Siamese neural network is used to identify individuals based on their rMEG FC metrics. As seen in **Figure 7**, there are three main components needed for building a Siamese network:

- Two identical ***ANNs***, also called the embedded networks, which share the same parameters. Design of the ANN is covered in section 2.2.1.
- A ***distance function*** that computes the difference between the output feature vectors of the twin ANNs. Following the design of (Dey et al., 2017), the Euclidean distance measure is used. The Euclidean distance d is simply computed as the square root of the sum of the squared differences between the two vectors (a, b) (Dokmanic et al., 2015):

$$d = f(a, b) = \sqrt{\sum_{i} (a_i - b_i)^2} \tag{13}$$

- Lastly, a cost function is used to differentiate between the input pairs and evaluate similarity. The ***contrastive loss*** function is represented by equation (14) (Gildenblat & Klaiman, 2019; Koch et al., 2015).

$$C(d, y) = yd^2 + (1 - y) \times max(0, (margin - d))^2 \tag{14}$$

, where:

- y is the label. The label is equal to 1 for image pairs of the same class, otherwise, the label is 0.
- d is the Euclidean distance between the outputs of the twin ANNs.
- The margin value is greater than 0. Having a margin indicates that dissimilar pairs that are beyond this margin will not contribute to the loss.
- For dissimilar pairs, the loss becomes $C(d, 0) = max(0, (margin - d))^2$. To minimize the loss, d has to be maximized.
- For similar pairs, the loss becomes $C(d, 1) = d^2$. To minimize the loss, d has to be minimized.

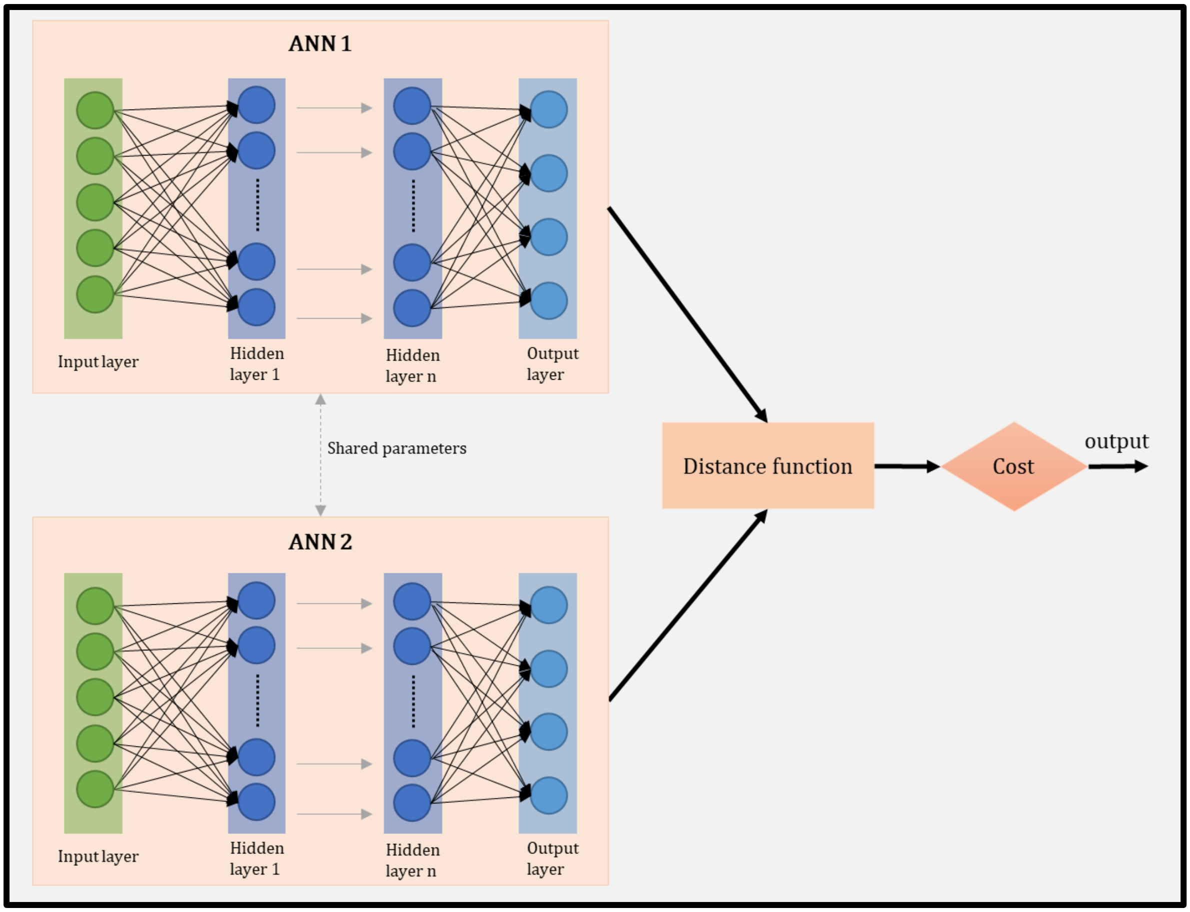


Figure 7: A simple representation of a Siamese Neural Network

2.2.3 Theoretical Background: Support Vector Machine

As mentioned earlier, the performance of the deep learning model will be compared against a machine learning technique, which is Support Vector Machine (SVM), proposed by (Cortes & Vapnik, 1995). SVM is a discriminative supervised machine learning algorithm that is able to construct a hyperplane or a set of hyperplanes in high-dimensional spaces that are usually used to discriminate between different classes (Noble, 2006). For example, SVM is used for classifying human actions from video footage (Schuldt et al., 2004), and text classification (Sun et al., 2009).

SVM can be divided into linear and nonlinear models depending on the data. The general steps for nonlinear SVM are: (1) mapping the data to a feature space using a nonlinear kernel function, (2) mapping the feature space into the response set, and (3) dividing the data into classes. Unlike neural networks, the SVM feature space depends on the kernel function rather than the number of hidden layers (Noble, 2006). The formula to describe SVM is denoted by equation (15), where W and b are the weights and biases and $\emptyset$ is the kernel function (Suthaharan, 2016):

$$y = W\emptyset x + b \tag{15}$$

Let the kernel function $\emptyset$ be represented as a function that maps two n-dimensional vectors (x, z) to a real number. Kernel functions can be in form of a linear function (equation 16), polynomial (equation 17), Gaussian Radial Basis Function (RBF) (equation 18), or sigmoid hyperbolic tangent (equation 19) (Patle & Chouhan, 2013):

$$x^T z \tag{16}$$

$$(x^T z + 1)^d \tag{17}$$

$$exp(-\gamma \|x - z\|^2) \tag{18}$$

$$\tanh(\gamma(x^T z) + c) \tag{19}$$

, where the parameters of these kernel functions are defined by the user:

- d denotes the degree and c is a constant. Both are arbitrary
- γ varies from 0 to 1

2.3 Literature Review

Deep Learning Applications in Neuroscience

Deep generative models are advantageous over machine learning mainly because of the following characteristics: (i) multiple layers can be greedily learned one layer at a time, (ii) greedy learning can be unsupervised, (iii) the ability to capture nonlinear distributed representations, and lastly (iv) performance can be improved further by fine-tuning the parameters of the final model (Salakhutdinov, 2015). These characteristics have allowed for deep learning methods to enhance the analysis for classification and representation learning using brain imaging. For example, in (Plis et al., 2014), researchers aimed to validate the feasibility of deep generative models, examine the effect of the depth in deep learning analysis, and determine the value of these methods for discovery of latent structure of a large dataset consisting of functional and structural magnetic resonance imaging (fMRI and sMRI). The results affirmed that deep learning models have a high potential in neuroimaging applications when compared to classical models used in the field.

Moreover, applying a deep learning approach to process and learn from neuroimages has been consistent in providing superior performance over uni and multivariate analysis. In the work of (Thomas et al., 2019), researchers have extensively reviewed multiple researches involving the usage of a deep learning approach, a CNN to be specific, to examine neurological disorders. The results were observed to be consistently successful and highly accurate. While this review focused on the neurologically ill population, the same can be inferred for the healthy population.

In (X. Li et al., 2020), researchers proposed a CNN model to process the EEG signals for classifying healthy and mildly depressed patients. The CNN's input was of an image type with 3 RGB channels (Red Green and Blue color channels). These images were transformed from the functional connectivity matrices that were calculated for every frequency band in the EEG (delta, theta, alpha, beta, and gamma band). Another research introduced a deep learning convolutional neural network (DCNN) workflow to process a spatiotemporal feature space (Hasasneh et al., 2018). That workflow was designed to automatically classify and remove unwanted ocular and cardiac artifacts form MEG/EEG data without the need for a reference, such as EOG and ECG recordings.

While (Hasasneh et al., 2018; C. F. Liu et al., 2019; Plis et al., 2014; Thomas et al., 2019) and many others have provided valuable insights affirming the superiority of deep learning models over conventional methodologies for pattern recognition tasks, the case of individual identification has not been considered.

Deep Similarity Learning

Measuring similarity between two lists of elements has been, for many years, a fundamental goal in computer science research (W. Liu et al., 2019). There are many approaches to compute similarity, each is suitable for specific contexts. For example, the Euclidean distance is mainly used to highlight the geometric differences between two elements (Liberti & Lavor, 2017), and Spearman's coefficient is used when the rank order correlation is the focus of the study (Chengwei et al., 2015).

All known measures of similarity have only been suitable for data samples which have the same dimensionality and type, that is until Bromley et al. introduced the ***Siamese Neural***

Network in 1993 to solve an image matching problem for handwritten signature verification (Bromley et al., 1993). The complexity of the data was caused due to the characteristics of collected signatures, where each sample had different features even if it came from the same subject. For example, handwritten signatures of the same person would differ in the dimension and position of each character, the space occupied for the signature, the speed of signing, and others. The proposed architecture for the Siamese Neural Network consisted of a couple of identical artificial neural networks that work synchronously in learning features from distinct inputs and then compare their outputs through the cosine distance measure to provide a similarity result. Eventually, that network was able to successfully indicate if the two signatures were genuine, or if one was a forgery. (Bromley et al., 1993; Chicco, 2021).

Ever since the appearance of the Siamese neural network model, it has been widely employed in various domains. An example from the medical imaging domain is (C. F. Liu et al., 2019), where the researchers proposed a deep learning framework based on the Siamese neural networks to detect neuroanatomical volume and shape asymmetries related to Alzheimer's disease from whole brain Magneto Resonance Imaging (MRI) data.

Furthermore, research in the signature forgery detection domain has been thriving after Bromley et al. published their novel approach (Bromley et al., 1993). Some of the recent examples are (Dey et al., 2017; Grafilon et al., 2017). The researchers of (Grafilon et al., 2017) developed a mobile application to detect handwritten signature forgeries from photos taken by the smart phone's camera. The detection system was based on the Siamese neural network design. As for (Dey et al., 2017), researchers have developed a robust convolutional Siamese neural network model named '*SigNet*' for signature verification capable of handling forgery detection in different languages and handwriting styles. ‘*SigNet’* model used the Euclidean distance measure for comparison between the output of the twin neural nets. That approach is considered unconventional compared to many studies that followed the Bromley et al.’s original design using the cosine similarity distance measure.

One of the most popular papers is by the name *Siamese Neural Networks for One-Shot Image Recognition* (Koch et al., 2015). The researchers have developed an impressive robust convolutional neural network framework based on the Siamese nets that successfully recognizes similar images using a single training sample per category. That work has proved the advantage of deep similarity learning over the traditional discriminative neural network approach for cases where the number of target categories is large and the number of training samples per category is very small.

2.4 Summary

MEG data is a considered a rich source for neuroscientists to infer descriptive features of the brain function. Two of these descriptive features are used for this study: Coherence and Amplitude Envelope Correlation. These features, also referred to as the functional connectome metrics of the subjects can be used to identify individuals from a cohort. As previously mentioned, many scientists have succeeded in creating statistical and machine learning models that performed robustly in identifying individuals. However, our goal for this study is to introduce a novel approach in this domain, which is deep similarity learning. Siamese neural networks have proven over time to be advantageous over commonly known designs using convolutional neural networks for image recognition when the data is fairly small compared to the number of labels to predict. Lastly, a nonlinear SVM model will be used to benchmark the results of machine learning against deep learning.

DEEP LEARNING MODEL DESIGN

3.1 Introduction

This chapter includes details about the workflow for this research. The FC data is processed according to the pipelines described in section 3.3, and then fed into the deep learning model, which is based on the Siamese neural network. Training of the Siamese network and the most promising design are defined in sections 3.4.1 and 3.4.2, respectively.

3.2 About the Dataset

The Human Connectome Project (HCP) represents the first attempt to collect and publicly share sufficient data on a large scale for the purpose of addressing the fundamental research questions about human connectional anatomy and variation (Van Essen et al., 2013). The data is based on an open dataset provided by the HCP named “1200 Subjects Data Release”. To describe the data in brief, the dataset includes high quality neuroimages from 1206 healthy young adults collected from August 2012–October 2015. The rMEG scans only were used from this dataset which were acquired from 95 subjects in 3 sessions with an approximate duration of 6 minutes for each session. To acquire the rMEG scans, subjects were kept comfortable and were instructed to remain still and relax with eyes open during scanning (*WU-Minn HCP 1200 Subjects Data Release Reference Manual*, 2018).

3.3 Data Processing

A general view for the data processing procedures is illustrated in **Figure 8**, where the raw rMEG data is fed into the HCP preprocessing pipeline, and the resulting FC data is further processed and transformed. Each process is explained in greater details below.

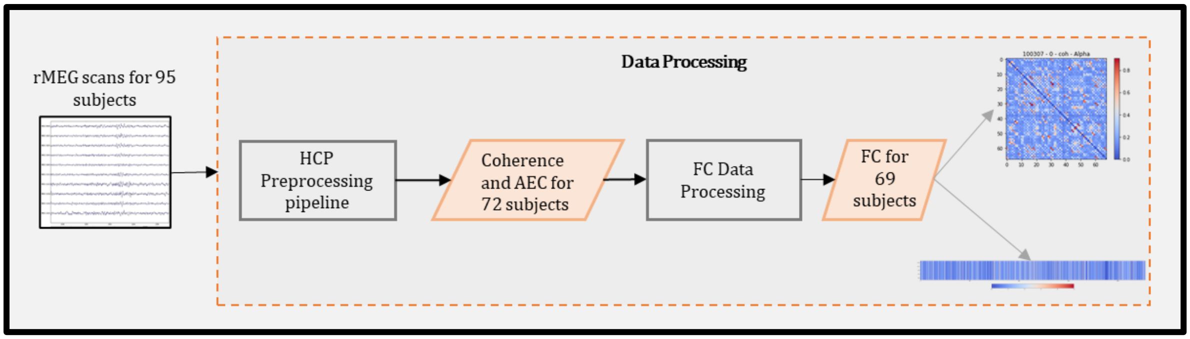


Figure 8: Data Processing Block Diagram. The rMEG scans from the HCP are preprocessed to compute the coherence and AEC. The FC data is then further processed to be used in proper format as input for subsequent machine and deep learning analysis

The ***HCP preprocessing pipeline*** includes noise artifact removal and sensor level analyses (Larson-Prior et al., 2014). Removal of data with larger noise and missing entries resulted in a decrease in the number of subjects used for analysis from 95 to 72. After time resolved source localization (Gramfort et al., 2014), the connectivity values between time courses of neuronal activity from 68 brain regions were computed. The connectivity metrics chosen for this research are coherence and AEC.

The rMEG FC measures are affected by the selection of the frequency bins and it is common practice to compute connectivity measures over the known classical frequency bands Delta, Theta, Alpha, Beta, and Gamma (Papanicolaou et al., 2020). A widely used definition of the frequency bands is adapted for this research as follows:

- Delta (1– 4) Hz, Theta (4– 8) Hz, Alpha (8– 13) Hz, Low Beta (13– 18) Hz, High Beta (18-30) Hz, Low Gamma (30– 80) Hz, and High Gamma (80– 200) Hz.

The next step is further processing this FC data and transforming into proper shape for analysis and modelling. The ***FC Data Processing*** procedure begins with further cleaning the data to remove inconsistencies across subjects. For example, certain subjects had been through 2 rMEG sessions instead of 3, therefore, data for these subjects was eliminated. As a result, the number of subjects used for analysis was decreased to 69 subjects with equal number of resting state recordings among all subjects. Regardless of its impact on the volume of data, the data cleaning task is vital to make sure that “dirty” data does not cause erroneous analysis and predictions (Chu et al., 2016; Rahm & Do, 2000).

Afterwards, this clean FC data is transformed by applying local normalization. Two techniques were used for comparison purposes: Min-max and Z-score normalization. Min-max normalization is a linear transformation technique to normalize data, where the minimum and maximum values of a data range are fetched, and all values in this range are transformed according to equation (20). Z-score normalization is used for standardizing data on the same scale according to equation (21). It is a measure for the number of standard deviations σ by which a given data point is above or below of the mean μ. (Patro & Sahu, 2015; Quackenbush, 2002).

$$x' = \frac{x - \min\,(\mathrm{x})}{\max\,(x) - \min\,(x)} \tag{20}$$

$$z = \frac{x - \mu}{\sigma} \tag{21}$$

The processed FC values remained in vectorized form for the SVM but were transformed into square shape matrices (resembling 2-dimensional images) for the Siamese neural network. Each vector or image represents FC values for 1 subject, over 1 frequency band, for 1 metric (Coherence or AEC), and for 1 experimental run. The vectors and matrices are displayed with colors in this figure for visualization purposes only.

3.4 The Siamese Neural Network

The following sections present the training method and architecture of the Siamese network. It is worth noting that the Keras library (Chollet, 2015) was used for the implementation.

3.4.1 Training the Neural Network

The FC images were grouped, where every image had 2 matching pairs and 2 non-matching pairs. Afterwards, following common practice (Xu & Goodacre, 2018), the image pairs were split into 3 sets: 60 % for training, 20% for validation, and 20% testing. The flowchart for training the Siamese neural network is shown in **Figure 9**.

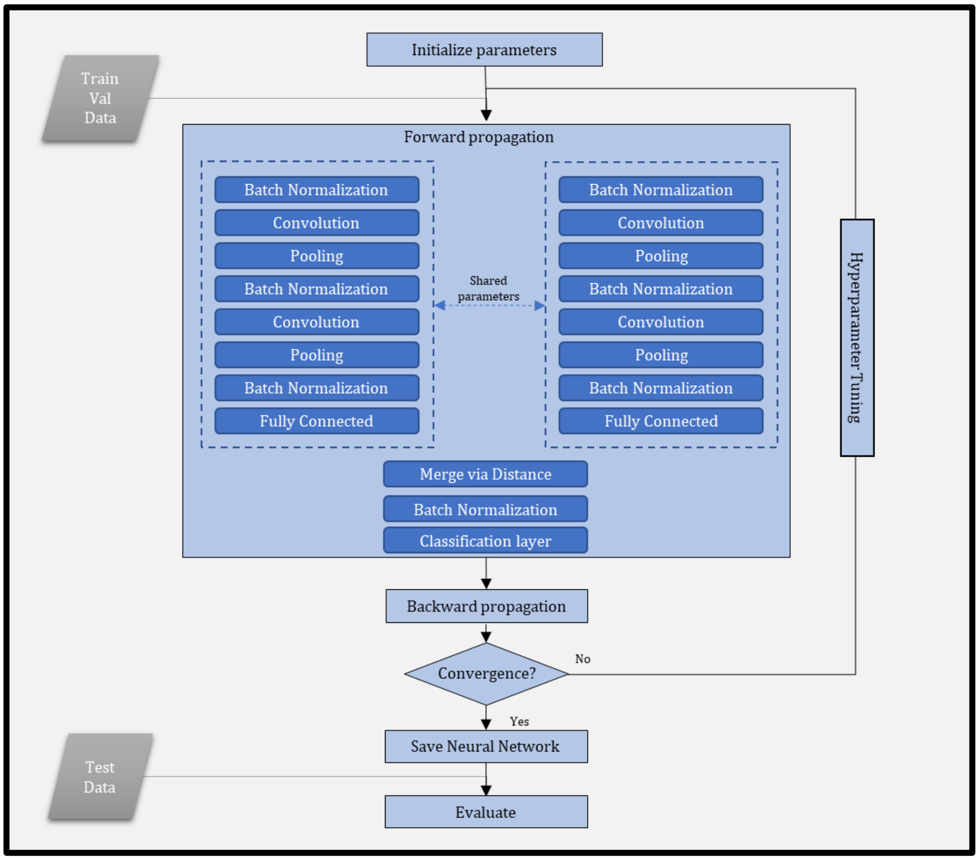


Figure 9: Flowchart for training the Siamese Neural Network

The training starts with initialized weights and biases for the input layer by using the normalization method by (He et al., 2015). Batch normalization layers are added before every activation function to normalize intermediate weights and biases with a zero mean and a unit variance. Lastly, the hyperparameters for the embedded CNNs are tuned manually, such that parameters like the kernel size, the kernel numbers, the pooling type, the pooling size, the learning rate, and others are tested empirically while monitoring the loss and accuracy values over the validation set.

3.4.2 Most Promising Architecture

As discussed in the previous section, the architecture for the Siamese network was defined via empirical testing, where the parameters for the embedded CNN are:

- First convolutional layer has 8 kernels
- Second convolutional layer has 16 kernels
- The kernels for both convolutional layers are of size (3,3) and have a stride of (1,1)
- Average Pooling is used with a pool size of (2,2)
- ReLu activation function is used for hidden layers
- Sigmoid activation function is used for the classification layer
- Adam Optimizer is used with a learning rate of 0.001

The resulting architecture of the Siamese neural network is shown in **Figure 10**, where:

- The **input data** is a 2-dimensional symmetric array of size (68 x 68) that is fed into a single channel of the CNN as a grayscale image.
- Each **embedded CNN network** consists of 2 hidden layers followed by fully connected layers.

- The Euclidean distance function acts as the **merging layer** for the CNN outputs.
- **Contrastive loss** is used to compute the output which is a binary label representing the semantic similarity between the projected representation of the input FC images.

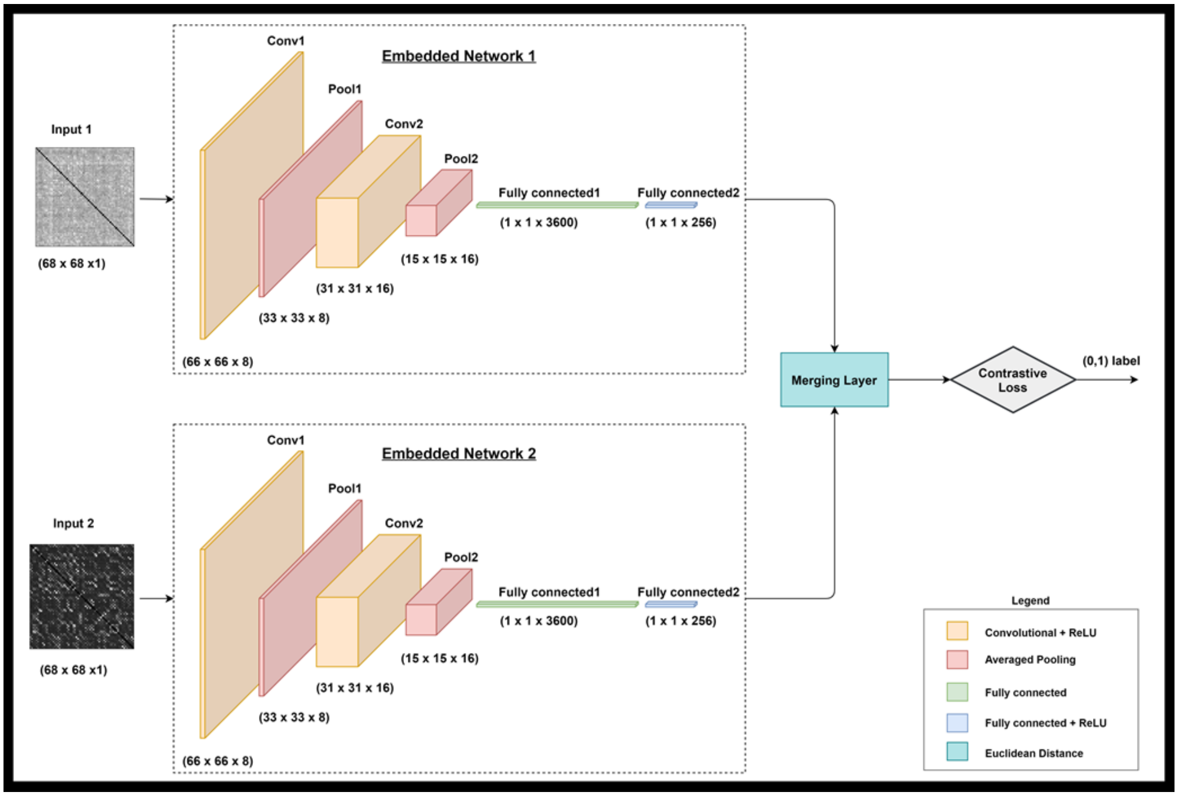


Figure 10: FC image pairs are fed into the twin CNNs. Each CNN has 2 hidden layers of convolution followed by Averaged pooling and then two fully connected layers. Outputs from the CNNs are merged via the Euclidean distance and then contrastive loss gives the similarity result

3.5 Summary

The rMEG Data is provided from the HCP and then processed, cleaned, and transformed into FC vectors and matrices. The processing procedures reduced the amount of data, but they remain vital for ensuring no “dirty” data is fed into the learning algorithms. The FC vectors and matrices included coherence and AEC measures over 7 different frequency bands. The SVM model requires data to be in vectorized form, and the Siamese network requires the matrix-data to be fed in pairs. Data pairs were created by grouping matching and non-matching pairs. Lastly, the most promising design of the Siamese network was defined via empirical testing of the parameters depending on the accuracy and loss of the validation dataset.

RESULTS AND DISCUSSION

4.1 Feature extraction

It is important to note down the amount of FC data that is available for learning. There are 69 subjects, where each subject has a total of 42 matrices or vectors for the machine learning and deep learning models to use for recognizing the patterns inside. The number 42 is acquired from 3 different rMEG experiments and 2 connectivity metrics computed for 7 different frequency bands.

Also, it is important to understand and compare the impact of the normalization onto the FC data. **Figure 11** shows a side-to-side comparison of normalized and unnormalized FC data. As mentioned previously, coherence and AEC originally range from 0 to 1, refer to section 2.1.1. This explains why Min-max normalization did not alter the data by much, where Z-score standardization transforms the data distribution as expected. The FC matrices with the Z-scores show higher contrast and lower brightness.

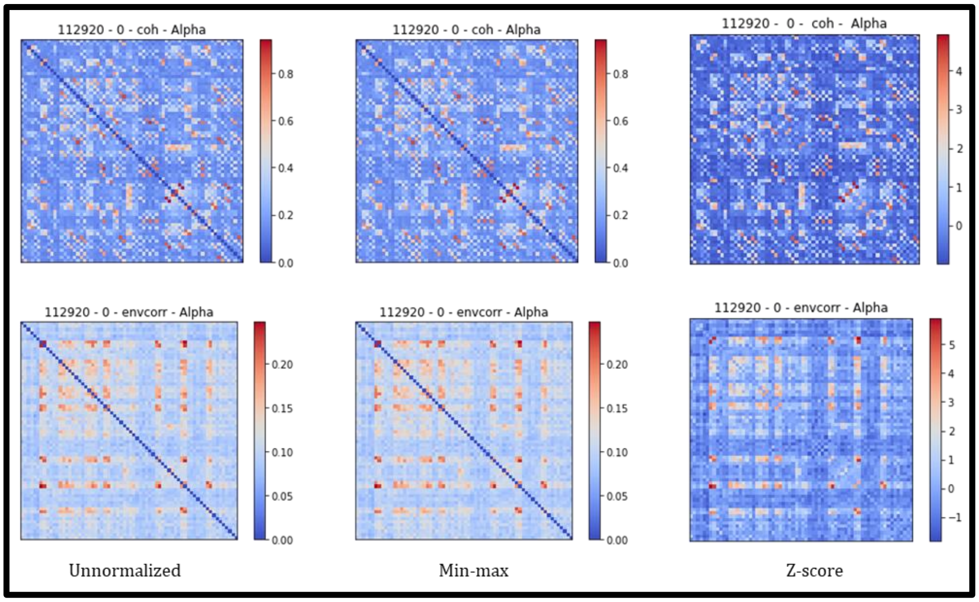
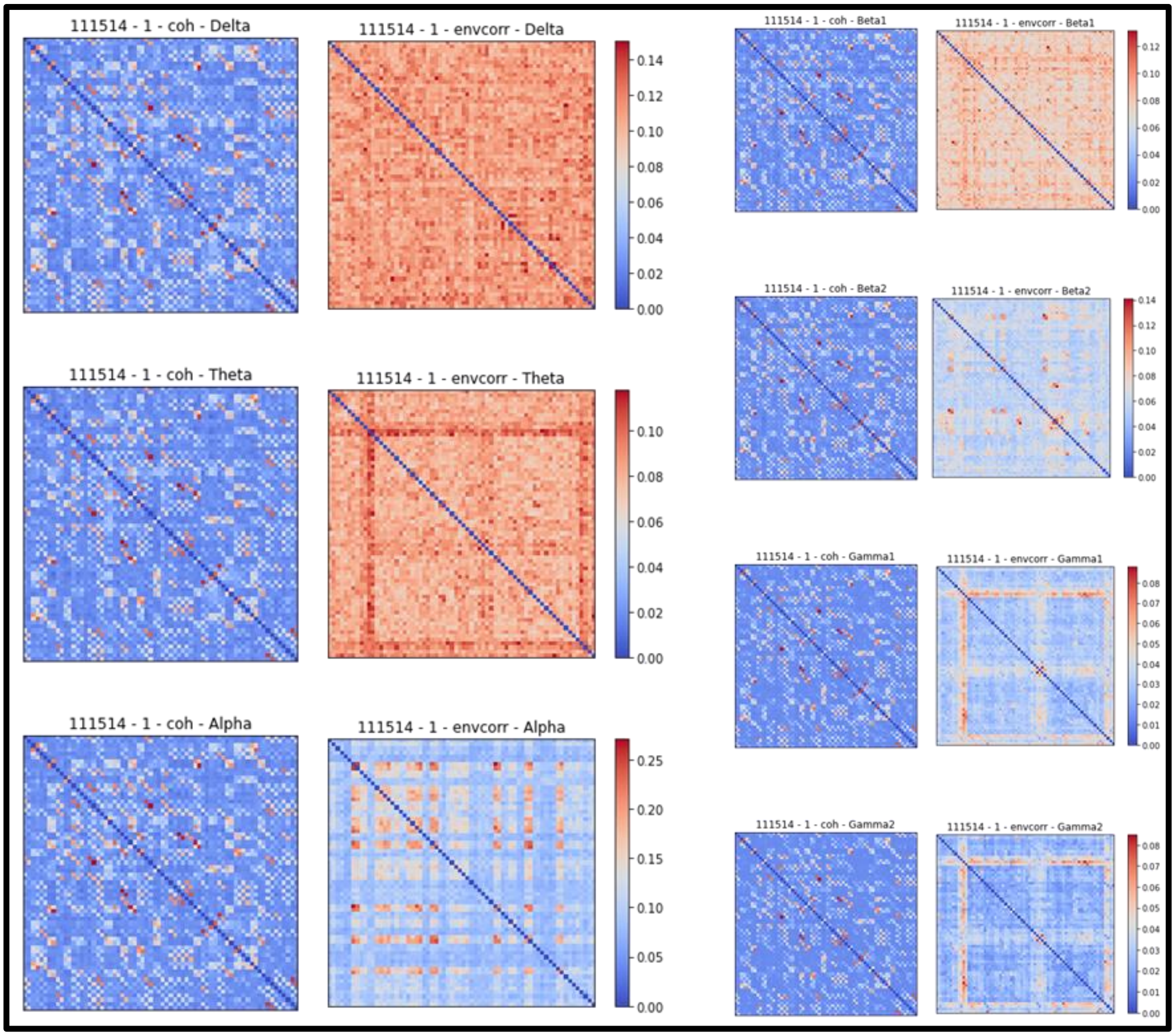


Figure 11: Comparison of normalized and unnormalized FC data

Furthermore, it is important to investigate the differences of the FC measures over the 7 frequency bands. **Figure 12** shows a side-to-side comparison of the coherence and AEC FC measures for 1 subject for each frequency band. It is observed, with the help of the color map, that AEC measure is impacted more than coherence by the frequency ranges. Another observation is that the 42 FC metrics per subjects do not necessarily include similar patterns, which increases the level of complexity for learning.



111514 - 1 - coh - Delta
111514 - 1 - envcorr - Delta
111514 - 1 - coh - Theta
111514 - 1 - envcorr - Theta
111514 - 1 - coh - Alpha
111514 - 1 - envcorr - Alpha
111514 - 1 - coh - Beta1
111514 - 1 - envcorr - Beta1
111514 - 1 - coh - Beta2
111514 - 1 - envcorr - Beta2
111514 - 1 - coh - Gamma1
111514 - 1 - envcorr - Gamma1
111514 - 1 - coh - Gamma2
111514 - 1 - envcorr - Gamma2

Figure 12: Min-max normalized FC values over the 7 frequency bands for a single subject

4.2 Siamese Neural Network Model Results

The Siamese neural network was tested with normalized and unnormalized data to investigate the effectiveness of the normalization methods. It can be seen in **Table 3** that using normalization helps with increase the accuracy and decrease the loss. As for normalization, Z-score method provided slightly better results than Min-Max with an impressive accuracy of 97.4% and a loss of 2.2%.

Table 3: Siamese Network Results showing prediction accuracy and loss for normalized and unnormalized data

Normalization	Accuracy	Loss
Min-max	0.9638	0.0291
Z-score	**0.9741**	**0.0221**
None	0.9586	0.0319

The neural network's evaluation metrics which are accuracy of how often the predictions equal the labels and loss which is contrastive loss, are plotted in **Figure 13** for all of the test runs of the Siamese Network. The common observation for all tests is that the network converges by the 20th epoch or earlier. Nonetheless, using Z-score normalized data allows for faster convergence amongst other tested data. Additionally, the results for unnormalized and Min-max normalized data are very similar, which is expected because the data is slightly changes by the Min-max function. Also, it is observed that the model with Z-score normalized data has a smoother curve with fewer spikes. Spikes are usually reduced via regularization methods, so it is interesting to observe that Z-score normalization helped slightly regularize the performance. It should be noted that very high accuracy was observed from the first tests to tune the parameters of the Siamese Network. Adjusting the parameters for each trial resulted in an increased accuracy and reduced loss by a small value of 10^{-2} order.

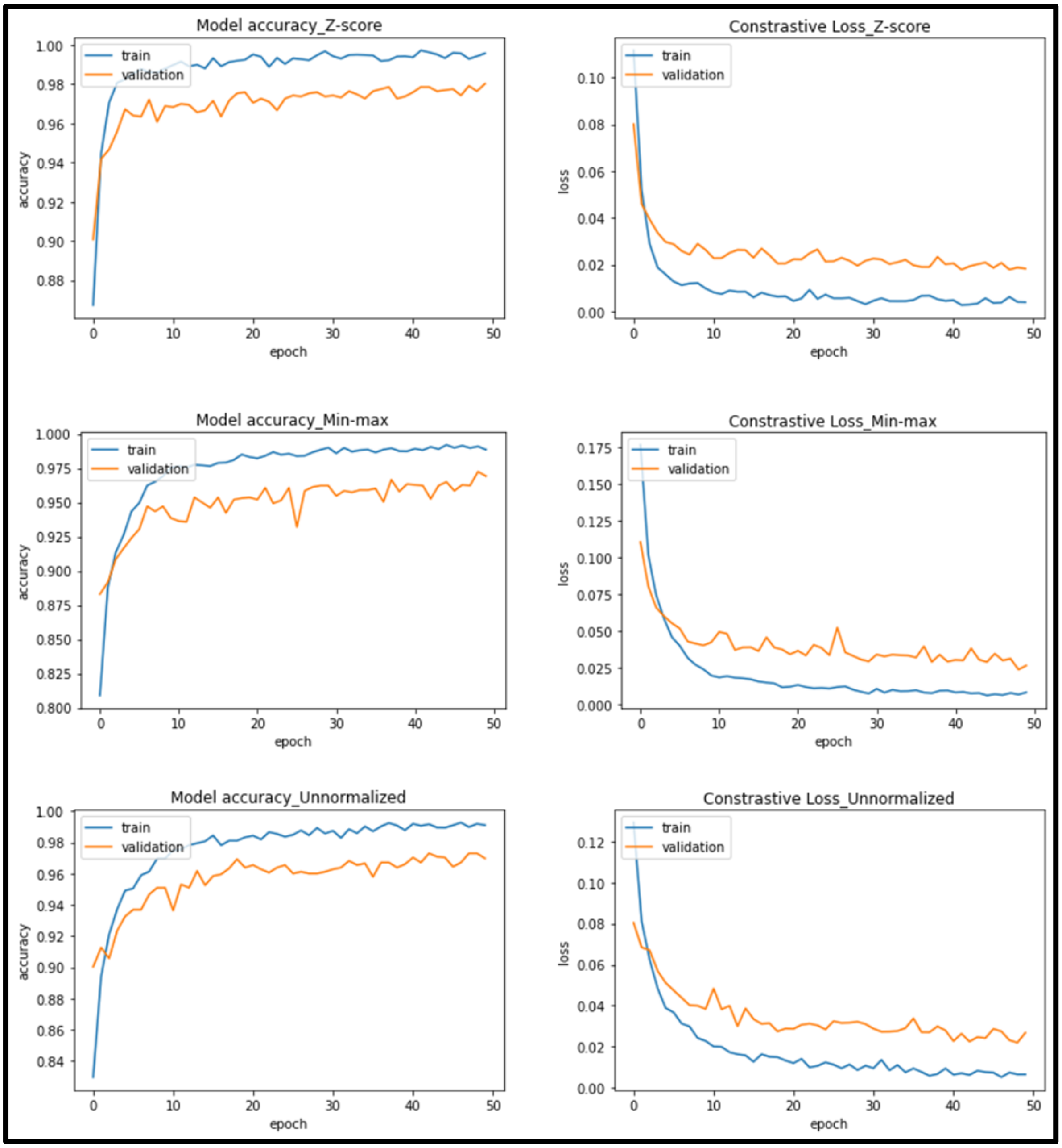


Figure 13: Siamese Network evaluation metrics for data normalized and unnormalized data

4.3 Support Vector Machine Results

SVM was tested with normalized and unnormalized FC vectors using 3 different types of nonlinear kernels, namely, polynomial, RBF, and sigmoid tanh. The results are displayed in **Table 4**, where:

- Precision represents the ratio of (true positives)/ (predicted positives)
- Recall represents the ratio of (true positives)/ (actual positives)
- The F1-score is the weighted average of the precision and recall

Table 4: SVM Results showing precision, recall, F1-score, and accuracy for normalized and unnormalized data

Kernel	Normalization	Parameter[1]	Precision	Recall	F1score	Accuracy
Polynomial	**Min-max**	***d* =2**	**0.84**	**0.81**	**0.81**	**0.81**
RBF	Min-max	γ =0.1	0.92	0.69	0.77	0.69
Sigmoid	Min-max	*c* =1, γ =default[2]	0.75	0.21	0.29	0.21
Polynomial	**Z-score**	***d* =2**	**0.84**	**0.80**	**0.81**	**0.80**
RBF	Z-score	γ =0.01	0.96	0.45	0.60	0.45
Sigmoid	Z-score	*c* =3, γ =default	0.82	0.78	0.78	0.78
Polynomial	None	*d* =2	0.93	0.47	0.60	0.47
RBF	None	γ =0.1	0.88	0.50	0.61	0.50
Sigmoid	None	*c* =3,γ =0.01	0.26	0.07	0.08	0.07

---

[1] Refer to section 2.2.3 for details about the parameters

[2] Default value is 1/(number of features*variance)

It is interesting to observe that the SVM did not perform well with unnormalized data, regardless of its high similarity to the min-max normalized data. As for normalized data, the most promising performance of the SVM is obtained by using either Z-score or Min-max normalized data with the polynomial kernel of the 2^{nd} degree. The polynomial SVM for normalized FC data was able to provide an 81% accuracy for Min-max normalized data with a high F1-score (0.8) indicating that the SVM had a very good ratio of true positive predictions compared to actual positive predictions.

The RBF SVM kernel performed poorly across all trials even though it computes a similarity measure. Also, the sigmoid kernel performed the worst on all trials, and it was very challenging to tune its performance. The tanh sigmoid output can be interpreted a probability from [-1,1] which makes it a better fit for neural network classification tasks.

4.4 Summary

The data that is available for learning is relatively small compared to the huge amount of data that is usually used for deep learning. The data is not only small, but also highly complex, where the very few FC vectors of matrices available per subject are not necessarily similar. Regardless of these challenges for learning, the Siamese network was able to easily provide an impressive 97.4% prediction accuracy with a great convergence rate, where the SVM fell behind with 81% prediction accuracy. Furthermore, the Siamese network was able to robustly provide highly accurate predictions regardless of the method for normalizing or not normalizing the data, unlike the SVM whose performance was highly impacted by normalizing the FC data.

CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

The obtained results align with the initial assumption that the deep learning model, with special focus on CNNs, would outperform a classical computational model. Additionally, using the CNNs as commonly known to perform classification of 69 subjects with very few data is very challenging. However, using the CNN in such a design that computes the semantic similarity of the hidden features representation, provides highly impressive results, without the need to augment the data. Our results for the Siamese neural network affirmed that deep similarity learning is the most promising option for image pattern recognition when data is scarce.

5.2 Contribution and Possible Applications

This research introduced using a deep learning and machine learning model for the identification of neuronal biomarkers based on the brain functional connectivity. The next step is to work on publishing a paper in a scientific journal to share the work with the research community. Additionally, it would be beneficial to deploy deep learning-based software applications into the domain of neurophysiological diagnosis. Such applications would help specialists by saving the time needed to analyze the neuroimages and extract insights from them.

5.3 Future Work

There are many steps to build upon this research. Firstly, I would like to use other FC metrics to augment the data such as Phase Lag Index PLI, Phase Locking Value PLV, or others. Additionally, it would be interesting to study the impact of each frequency band on its own on the performance. To be specific, the FC images can be fed into 3 channels to the CNN, where the 3rd dimension would be used for frequency. Lastly, it would be a great collaboration opportunity to introduce the deep learning model into multi-modal FC identification, where multiple neuroimage types are used such as fMRI and EEG for the same subject.

5.4 Closing Remarks

Technological advancements in the neuroimaging domain have allowed for a cross over with the artificial intelligence applications, especially with the neuronal data becoming more openly available for researchers to explore. While it could be challenging at first to grasp the concepts behind neural signal processing, with proper guidance from domain experts, a data scientist with pure engineering background like myself would be able to manipulate the data like any other and conduct proper research in machine learning and deep learning to solve the field specific problem.

BIBLIOGRAPHY

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Alom, Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Hasan, M., Van Essesn, B. C., Awwal, A. A. S., & Asari, V. K. (2019). A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, *8*. https://doi.org/10.3390/electronics8030292

Amico, E., & Goñi, J. (2018). *The quest for identifiability in human functional connectomes*. *January*, 1–14. https://doi.org/10.1038/s41598-018-25089-1

Artificial Neural Networks. (2021). In H. Cartwright (Ed.), *Methods in Molecular Biology* (Third, Vol. 2190). Humana. https://doi.org/https://doi.org/10.1007/978-1-0716-0826-5

Barcha, D. M., Burgess, G. C., Harms, M. P., Petersena, S. E., Schlaggar, B. L., Corbetta, M., Glasser, M. F., Curtiss, S., Dixit, S., Feldt, C., Nolan, D., Bryant, E., Hartley, T., Footer, O., Bjork, J. M., Poldrack, R., Smith, S., Johansen-Berg, H., Snyder, A. Z., & Van Essen, D. C. (2014). *Function in the Human Connectome: Task-fMRI and Individual Differences in Behavior*. 169–189. https://doi.org/10.1016/j.neuroimage.2013.05.033.Function

Basu, J. K., Bhattacharyya, D., & Kim, T. (2010). Use of Artificial Neural Network in Pattern Recognition. *International Journal of Software Engineering and Its Applications*, *4*(2), 23–34.

Bock, C., Moor, M., Jutzeler, C. R., & Borgwardt, K. (2021). Machine Learning for Biomedical Time Series Classification: From Shapelets to Deep Learning. In H. Cartwright (Ed.), *Artificial Neural Networks. Methods in Molecular Biology. vol2190* (3rd Editio, pp. 33–60). https://doi.org/https://doi.org/10.1007/978-1-0716-0826-5_2

Bowyer, S. M. (2016). Coherence a measure of the brain networks: past and present. *Neuropsychiatric Electrophysiology*, *2*(1), 1–12. https://doi.org/10.1186/s40810-015-0015-7

Bromley, J., Bentz, J. W., Bottu, L., Guyon, I., Lecun, Y., Moore, C., Sackinger, E., & Shah, R. (1993). Signature verification using a"Siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, *07*(04), 669–688. https://doi.org/10.1142/s0218001493000339

Brookes, M. J., Hale, J. R., Zumer, J. M., Stevenson, C. M., Francis, S. T., Barnes, G. R., Owen, J. P., Morris, P. G., & Nagarajan, S. S. (2011). Measuring functional connectivity using MEG: Methodology and comparison with fcMRI. *NeuroImage*, *56*(3), 1082–1104. https://doi.org/10.1016/j.neuroimage.2011.02.054

Chengwei, X., Jiaqi, Y., Rui Máximo, E., & Rong, C. (2015). Using Spearman’s correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience*. https://doi.org/10.1002/cpe.3745

Chicco, D. (2021). Siamese Neural Networks: An Overview. In H. Cartwright (Ed.), *Artificial Neural Networks. Methods in Molecular Biology. vol2190* (3rd Editio, pp. 73–94). Humana. https://doi.org/https://doi.org/10.1007/978-1-0716-0826-5_3

Chollet, F. (2015). Keras. *GitHub Repository*. https://github.com/fchollet/keras

Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data Cleaning: Overview and Emerging Challenges. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Vols. 26-June-20, pp. 2201–2206). Association for Computing Machinery. https://doi.org/10.1145/2882903.2912574

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Da, J., Castanheira, S., Orozco, H. D., Misic, B., & Baillet, S. (2021). *MEG , myself , and I : individual identification from neurophysiological brain activity*. 1–30.

Dahl, G., Sainath, T., & Hinton, G. (2013). Improving deep neural networks for LVCSR using rectified linear units and dropout. *2013 IEEE International Conference on Acoustics; Speech and Signal Processing*, 8609–8613. https://doi.org/10.1109/ICASSP.2013.6639346

Demuru, M., Gouw, A. A., Hillebrand, A., Stam, C. J., Dijk, B. W. Van, & Scheltens, P. (2017). *Functional and effective whole brain connectivity using magnetoencephalography to identify monozygotic twin pairs*. *August*, 1–11. https://doi.org/10.1038/s41598-017-10235-y

Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Llados, J., & Pal, U. (2017). SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. *ELSEVIER Pattern Recognition Letters*, *1*, 1–7. http://arxiv.org/abs/1707.02131

Dokmanic, I., Parhizkar, R., Ranieri, J., & Vetterli, M. (2015). Euclidean Distance Matrices: Essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, *32*(6), 12–30. https://doi.org/10.1109/MSP.2015.2398954

Filippi, M. (2015). *Oxford Textbook of Neuroimaging*. Oxford University Press.

Finn, E. S., Shen, X., Scheinost, D., Rosenberg, M. D., Huang, J., Chun, M. M., Papademetris, X., & Constable, R. T. (2015). Functional connectome fingerprinting : identifying individuals using patterns of brain connectivity. *Nature Publishing Group*, *18*(11), 1664–1671. https://doi.org/10.1038/nn.4135

Gerven, M. Van, & Bohte, S. (Eds.). (2018). *Artificial Neural Networks as Models of Neural Information Processing*. Lausanne: Frontiers Media. https://doi.org/10.3389/978-2-88945-401-3

Gildenblat, J., & Klaiman, E. (2019). *Self-Supervised Similarity Learning for Digital Pathology*. 1–9. http://arxiv.org/abs/1905.08139

Grafilon, P., Aguilar, I. B. S., Lavarias, E. D., Apalin, J. C. N., & Tan, F. V. (2017). A Signature Comparing Android Mobile Application Utilizing Feature Extracting Algorithms. *International Journal of Scientific & Technology Research*, *6*(08), 45–50.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., & Hämäläinen, M. S. (2014). MNE software for processing MEG and EEG data. *NeuroImage*, *86*, 446–460.

https://doi.org/10.1016/j.neuroimage.2013.10.027

Gross, J., Baillet, S., Barnes, G. R., Henson, R. N., Hillebrand, A., Jensen, O., Jerbi, K., Litvak, V., Maess, B., Oostenveld, R., Parkkonen, L., Taylor, J. R., van Wassenhove, V., Wibral, M., & Schoffelen, J. M. (2013). Good practice for conducting and reporting MEG research. *NeuroImage*, *65*, 349–363. https://doi.org/10.1016/j.neuroimage.2012.10.001

Günther, F., & Fritsch, S. (2010). Neuralnet: Training of neural networks. *R Journal*, *2*(1), 30–38. https://doi.org/10.32614/rj-2010-006

Hamalainen, M., Hari, R., Ilmoniemi, R. J., Knuutila, J., & Lounasmaa, O. V. (1993). Magnetoencephalography.Theory, instrumentation and applications to the noninvasive study of human brain function. *Rev. Mod. Phys*, *65*(2), 413--497. https://doi.org/10.1103/RevModPhys.65.413

Hasasneh, A., Kampel, N., Sripad, P., Shah, N. J., & Dammers, J. (2018). *Deep Learning Approach for Automatic Classification of Ocular and Cardiac Artifacts in MEG Data*. *2018*, 15–17.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers : Surpassing Human-Level Performance on ImageNet Classification. *IEEE International Conference on Computer Vision (ICCV)*, 1026–1034. https://doi.org/10.1109/ICCV.2015.123

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. 1–18. http://arxiv.org/abs/1207.0580

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML*, *1*, 448–456.

Japkowicz, N. (2001). Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, *42*(1–2), 97–122.

https://doi.org/10.1023/A:1007660820062

Karnin, E. D. (1990). A Simple Procedure for Pruning Back-Propagation Trained Neural Networks. *IEEE Transactions on Neural Networks*, *1*(2), 239–242. https://doi.org/10.1109/72.80236

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese Neural Networks for One-shot Image Recognition. *ICML Deep Learning Workshop*.

Kong, W., Wang, L., Xu, S., Babiloni, F., & Chen, H. (2019). EEG Fingerprints : Phase Synchronization of EEG Signals as Biomarker for Subject Identification. *IEEE Access*, *PP*, 1. https://doi.org/10.1109/ACCESS.2019.2931624

Koopmans, L. (1995). *The Spectral Analysis of Time Series* (Z. W. Birnbaum & E. Lukacs (Eds.); 1st editio). Academic Press.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (NIPS'12, pp. 1097–1105). Curran Associates Inc.

Larson-Prior, L. J., Oostenveld, R., Della Penna, S., Michalareas, G., Prior, F., Babajani-Feremi, A., Schoffelen, J.-M., Marzetti, L., de Pasquale, F., Di Pompeo, F., Stout, J., Woolrich, M., Luo, Q., Bucholz, R., Fries, P., Pizzella, V., Romani, G. L., Corbetta, M., & Snyder, A. Z. (2014). *Adding dynamics to the Human Connectome Project with MEG: Special NeuroImage issue on 'Mapping the Connectome.' 7386*, 190–201. https://doi.org/10.1016/j.neuroimage.2013.05.056.Adding

LeCun, Y. (1989). Generalization and network design strategies. *Technical Report CRG-TR-89-4*, 9.

Li, C., Mai, Y., Dong, M., Yin, Y., Hua, K., Fu, S., & Wu, Y. (2019). Multivariate Pattern Classification of Primary Insomnia Using Three Types of Functional Connectivity Features. *Frontiers in Neurology*, *10*(October), 1–10. https://doi.org/10.3389/fneur.2019.01037

Li, J., Cheng, J., Shi, J., & Huang, F. (2012). Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement. In D. Jin & S. Lin (Eds.), *Advances in Computer Science and Information Engineering. Advances in Intelligent and Soft Computing* (Vol. 169, pp. 553–558). Springer. https://doi.org/https://doi.org/10.1007/978-3-642-30223-7_87

Li, X., La, R., Wang, Y., Hu, B., & Zhang, X. (2020). A Deep Learning Approach for Mild Depression Recognition Based on Functional Connectivity Using Electroencephalography. *Frontiers in Neuroscience*, *14*(April), 1–20. https://doi.org/10.3389/fnins.2020.00192

Liang, Y., Liu, B., Li, X., & Wang, P. (2018). Multivariate Pattern Classification of Facial Expressions Based on Large-Scale Functional Connectivity. *Frontiers in Human Neuroscience*, *12*(March), 1–12. https://doi.org/10.3389/fnhum.2018.00094

Liberti, L., & Lavor, C. (2017). *Euclidean Distance Geometry: An Introduction*. Springer International Publishing. https://doi.org/10.1007/978-3-319-60792-4 Library

Liu, C. F., Padhy, S., Ramachandran, S., Wang, V. X., Efimov, A., Bernal, A., Shi, L., Vaillant, M., Ratnanather, J. T., Faria, A. V., Caffo, B., Albert, M., & Miller, M. I. (2019). Using deep Siamese neural networks for detection of brain asymmetries associated with Alzheimer’s Disease and Mild Cognitive Impairment. *Magnetic Resonance Imaging*, *64*(July), 190–199. https://doi.org/10.1016/j.mri.2019.07.003

Liu, W., Liu, Z., Rehg, J. M., & Song, L. (2019). Neural similarity learning. *Advances in Neural Information Processing Systems*, *32*(NeurIPS), 1–12.

Mash, R., Becherer, N., Woolley, B., & Pecarina, J. (2016). Toward aircraft recognition with convolutional neural networks. *EEE National Aerospace and Electronics*

Conference (NAECON) and Ohio Innovation Summit (OIS), 225–232. https://doi.org/10.1109/naecon.2016.7856803

Nielsen, M. A. (2015). Neural Networks and Deep Learning. In *Determination Press*.

Noble, W. S. (2006). What is a support vector machine ? *Nature Biotechnology*, *24*(12), 1565–1567. https://doi.org/https://doi.org/10.1038/nbt1206-1565

Papanicolaou, A. C., Roberts, T. P. L., & Wheless, J. W. (Eds.). (2020). *Fifty Years of Magnetoencephalography*. Oxford University Press.

Patle, A., & Chouhan, D. S. (2013). SVM kernel functions for classification. *2013 International Conference on Advances in Technology and Engineering, ICATE 2013*. https://doi.org/10.1109/ICAdTE.2013.6524743

Patro, S. G. K., & Sahu, K. K. (2015). Normalization: A Preprocessing Stage. *IARJSET*, 20–22. https://doi.org/10.17148/IARJSET.2015.2305

Plis, S. M., Hjelm, D. R., Salakhutdinov, R., Allen, E. A., Bockholt, H. J., Long, J. D., Johnson, H. J., Paulsen, J. S., & Turner, J. A. (2014). *Deep learning for neuroimaging : a validation study*. *8*(August), 1–11. https://doi.org/10.3389/fnins.2014.00229

Pratiwi, H., Windarto, A. P., Susliansyah, S., Aria, R. R., Susilowati, S., Rahayu, L. K., Fitriani, Y., Merdekawati, A., & Rahadjeng, I. R. (2020). Sigmoid Activation Function in Selecting the Best Model of Artificial Neural Networks. *Journal of Physics: Conference Series*. https://doi.org/10.1088/1742-6596/1471/1/012010

Quackenbush, J. (2002). Microarray data normalization and transformation. *Nature Genetics, 32*(4S), 496–501. https://doi.org/10.1038/ng1032

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, *23*(4), 3–13.

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. 1–14. http://arxiv.org/abs/1609.04747

Sakkalis, V. (2011). Review of advanced techniques for the estimation of brain connectivity measured with EEG/MEG. *Computers in Biology and Medicine*, *41*(12), 1110–1117. https://doi.org/10.1016/j.compbiomed.2011.06.020

Salakhutdinov, R. (2015). Learning Deep Generative Models. *Annual Review of Statistics and Its Application*. https://doi.org/10.1146/annurev-statistics-010814-020120

Sareen, E., Zahar, S., Ville, D. Van De, Gupta, A., Griffa, A., & Amico, E. (2021). Exploring MEG brain fingerprints: Evaluation, pitfalls, and interpretations. *NeuroImage*, *240*(February), 118331. https://doi.org/10.1016/j.neuroimage.2021.118331

Schuldt, C., Barbara, L., & Stockholm, S.-. (2004). Recognizing Human Actions : A Local SVM Approach. *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, *3*, 32–36. https://doi.org/10.1109/ICPR.2004.1334462

Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, *4*(12), 310–316. https://doi.org/10.33564/IJEAST.2020.v04i12.054

Singh, S. P. (2014). Magnetoencephalography: Basic principles. *Annals of Indian Academy of Neurology*, *17*(SUPPL. 1). https://doi.org/10.4103/0972-2327.128676

Sun, A., Lim, E. P., & Liu, Y. (2009). On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems*, *48*(1), 191–201. https://doi.org/10.1016/j.dss.2009.07.011

Suthaharan, S. (2016). Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification. . Integrated Series in Information Systems*. Springer. https://doi.org/10.1007/978-1-4899-7641-3_9

Thomas, A. W., Heekeren, H. R., & Müller, K. (2019). *Analyzing Neuroimaging Data Through Recurrent Deep Learning Models*. *13*(December), 1–18. https://doi.org/10.3389/fnins.2019.01321

Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E. J., Yacoub, E., & Ugurbil, K. (2013). The WU-Minn Human Connectome Project: An overview. *NeuroImage*, *80*, 62–79. https://doi.org/10.1016/j.neuroimage.2013.05.041

WU-Minn HCP 1200 Subjects Data Release Reference Manual (Vol. 2017, Issue March 2017). (2018).

Wu, J. (2017). Introduction to Convolutional Neural Networks. *Introduction to Convolutional Neural Networks*, 1–31. https://web.archive.org/web/20180928011532/https://cs.nju.edu.cn/wujx/teaching/15_CNN.pdf

Xu, Y., & Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

Zamm, A., Debener, S., Bauer, A. K. R., Bleichner, M. G., Demos, A. P., & Palmer, C. (2018). Amplitude envelope correlations measure synchronous cortical oscillations in performing musicians. *Annals of the New York Academy of Sciences*, *1423*(1), 251–263. https://doi.org/10.1111/nyas.13738

Zoumpourlis, G., Doumanoglou, A., Vretos, N., & Daras, P. (2017). Non-linear Convolution Filters for CNN-Based Learning. *2017 IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV.2017.510

ABBREVIATIONS

Table 5. Terminology and Definitions

Term	Definition
AE	Amplitude Envelope
AEC	Amplitude Envelope Correlation
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DCNN	Deep Learning Convolutional Neural Network
EC	Effective Connectivity or Effective Connectome
EEG	Electroencephalogram
EOG	Electrooculogram
FC	Functional Connectivity or Functional Connectome
fMRI	functional Magnetic Resonance Imaging
HCP	Human Connectome Project
Hz	Hertz
MEG	Magnetoencephalography
MLP	Multi-Layer Perceptron
MRI	Magneto Resonance Imaging
MSE	Mean Square Error
MVPA	Multivariate Pattern Analysis
MZ	Monozygotic

Term	Definition
PCA	Principal Component Analysis
PLI	Phase Lag Index
PLV	Phase Locking Value
PSD	Power Spectral Density
RBF	Gaussian Radial Basis Function
ReLu	Rectified Linear Unit
rMEG	resting-state Magnetoencephalography
sMRI	structural Magnetic Resonance Imaging
SVM	Support Vector Machine

ملخص الرسالة

يعد التعرف على المؤشرات الحيوية العصبية على المستوى الفردي هدفًا شاملاً في أبحاث علم الأعصاب .حيث تمكن علماء الأعصاب من تحديد الأفراد بناءً على اتصال الدماغ الوظيفي والهيكلية للدماغ كما يوضح من التصوير المغناطيسي للدماغ (MEG)، أو التصوير بالرنين المغناطيسي الوظيفي (fMRI)، أو التصوير الكهربائي للدماغ (EEG). يجدر بالذكر أن عملية تحديد الهوية الفردية يشار إليها أيضاً في الأبحاث على أنها تحديد للبصمات الدماغية الفردية. بالنسبة لهذا البحث، فقد تم استخدام بيانات التصوير المغناطيسي في حالة الراحة لأفراد أصحاء من قاعدة بيانات تم اتاحتها من قبل مشروع الاتصال البشري (HCP). يقدم هذا البحث نهجًا جديدًا تجاه تحديد بصمات الدماغ الفردية من خلال تطبيق نموذج تعلم التشابه العميق، ما يسمى بالشبكة العصبية السيامية التي تعتمد على الشبكات العصبية التلافيفية. يتم أيضاً مقارنة النتائج مع خوارزمية آلة المتجه الداعم بهدف إثبات تفوق نهج التعلم العميق. بالنسبة للبيانات المستخدمة للتعلم، فقد تم استخراج عوامل الترابط والترابط المغلف من صور الدماغ المغناطيسية كمقاييس لاتصال الدماغ الوظيفي. تشير النتائج لاحقاً بتحقق التوقعات على أن التعلم العميق يتفوق على تعلم الآلة. تمكنت الشبكة السيامية من الحصول بسهولة على دقة في التنبؤ تصل إلى 97٪ مقارنة بدقة 81٪ القادمة من خوارزميات آلة المتجه الدعم. تشير النتائج أيضاً إلى أن تصميم الشبكة السيامية يمكن من التعرف على الأنماط من البيانات برغم شُحّها، على عكس الخوارزميات الأخرى.

في الختام، تعد الشبكات العصبية التلافيفية أداة حسابية قوية للغاية يمكنها تعزيز التحليل أي مجال، مثل علم الأعصاب أو غيره. حتى الآن، لم يستفد مجمع البحث الخاص بالبصمات الدماغية من تقنيات التعلم العميق، حيث يمكن استخدام هذا البحث كخطوة أولى نحو تعزيز الأبحاث للبصمات الدماغية للأفراد من الاتصال الوظيفي.

A Convolutional Neural Network Framework for the Identification of Individual Neuronal
Biomarkers based on Functional Brain Connectivity using Magnetoencephalography
Farah Mohammad Waseem Abdellatif
2021