REAL-TIME CLASSIFICATION OF FNIRS DATA TO DEVELOP A BRAIN COMPUTER INTERFACE USING DEEP LEARNING

A Thesis

Submitted By

Shaik Mastanvali

For the award of the degree of

MASTER OF TECHNOLOGY, CLINICAL ENGINEERING

Jointly offered by

Indian Institute of Technology, Madras

Christian Medical College, Vellore

Sree Chitra Tirunal Institute for Medical Sciences and Technology, Trivandrum

Is evaluated and approved by

Dr. Sujesh Sreedharan
(Guide) Engineer-F
BMT Wing,
SCTIMST.

Dr. C. Kesavadas
(Examiner)
PROFESSOR OF RADIOLOGY,
SCTIMST,
THIRUVANANTHAPURAM.

THESIS CERTIFICATE

This is to certify that the thesis entitled **"Real-time classification of fNIRS data to develop a brain computer interface using deep learning"** submitted by **Shaik Mastanvali** to SCTIMST Trivandrum, for the award of the degree of **Master of Technology in Clinical Engineering** jointly offered by IIT Madras, CMC Vellore and SCTIMST Trivandrum is a bonafide record of research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Sujesh Sreedharan
Engineer-F
Department of Biomedical Technology
Sree Chitra Tirunal Institute for Medical Sciences and Technology
Trivandrum– 695012.

Place: Trivandrum

Date: June 2021

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my guide, Dr. Sujesh Sreedharan, to work with him. His approach has always inspired me and motivated me to learn more. His idea of reading any research paper or thesis helped me a lot to gain insights for the project. I would like to thank Project staff, Caleb Jones Shibu, for assisting me with my research every day. Also, I would like to thank Chandrasekharan Kesavadas and project staff in (Division of Artificial Internal Organs) for their help with my measurements. Without their cooperation and assistance in their lab, I would not be able to complete this work. I am also grateful for all the love, support, and encouragement that I received from my family and my friends without whom I would never have come so far in my life.

Shaik Mastanvali

BT19M024

ABSTRACT

Keywords: Brain state classification, functional near-infrared spectroscopy, brain-computer interface, deep learning, convolutional neural networks

This thesis presents the study of Real time classification of fNIRS data to develop a brain computer interface using deep learning technique. A 1D CNN deep learning classifier trained on fNIRS data from three healthy subjects is analyzed for real time performance in terms of classification accuracy during epochs of rest, left active, and right active hand movements. The purpose of this study is to optimize the labelling approach, number of layers of 1D CNN and frequency band of fNIRS data to improve the real time classification accuracy. The labelling approach based on the latest 2 seconds of data in the band 0.01-1Hz and 2 layers of CNN gave the best overall performance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

ABBREVIATIONS

fNIRS -Functional Near-Infrared Spectroscopy

BCI -Brain Computer Interface

CNN -Convolution Neural Network

ANN -Artificial Neural Network

MLFF -Multilayer Feed Forward layers

ReLU -Rectified Linear Activation function

TSS -Total Sum of squares

MSE - Mean Squared Error

RMSE - Root Mean Squared Error

PCA - Principal Component Analysis

ICA -Independent Component Analysis

HbO -Oxy hemoglobin

HbR -deox hemoglobin

HbT -Total hemoglobin

CHAPTER-1

INTRODUCTION

Background:

Functional Near-Infrared Spectroscopy

fNIRS stands for functional near-infrared spectroscopy and is a non-invasive optical imaging technique. For characterizing real-time brain oxygenation, This technique takes advantage of the optical differences between oxygenated and deoxygenated hemoglobin. fNIRS measures the concentration of hemoglobin(Hb) from change s in characteristic absorption of near infrared light.

Total six different possibilities for infrared light to interact with the brain tissue:

1. Direct transmission
2. Diffuse transmission
3. Specular reflection,
4. Diffuse reflection,
5. Scattering
6. Absorption.

fNIRS mainly focuses on absorption: differences in the absorption spectra of deoxy-Hb and oxy-Hb allow the measurement of relative changes in hemoglobin concentration using light attenuation at multiple wavelengths. Two wavelengths are selected, in which one is wavelength above the isosbestic point and one is below the isosbestic point. Isosbestic point (810nm) is defined at which deoxy-Hb and oxy-Hb have identical absorption coefficients.

Problem Statement:

The present work focuses on the developing a fnirs based brain computer interface with the help of 1D CNN classifier model for signal classification during epochs of rest, left active, and right active hand movements.

1. Identifying the best labelling approach for the classifier in a real time scenario.
2. Find the best number of convolution layers to be used in classifier to get good accuracy.
3. Identifying the Bandpass filter passband in between 0.01-0.2Hz 0.01 -1Hz to get best possible accuracy.

CHAPTER 2

Basics of 1D CNN

2.1 Introduction

An ANN is a structure made up of a large number of linked neurons. Each unit computed the input, which can also be the output of neurons from the previous layer. Backpropagation is the most important algorithm while training the ANN model. The ANN model consists of input layer, hidden layer(s) and an output layer. During forward propagation, the input to the neural network is transferred to the output using a loss function and optimization method. To decrease the loss function, the ANN's weights are updated every epoch.

ANN’s are good at solving some problems, but in some cases, they facing from two main challenges.

1. ANNs with fewer hidden layers are insufficient to tackle more complicated issues, necessitating the addition of more hidden layers to achieve the optimum results.
2. The second issue arises when choosing the optimal characteristics from a high-dimensional input such as photos. However, if the best features aren't extracted, the ANN is prone to overfitting the data. Extraction of the greatest potential features for a specific task is quite tough.

Convolutional neural networks easily overcome the second challenge since they handle the best feature selection and use fewer neurons because the weights are shared across the number of convolution layer outputs.

Convolutional Neural Networks (CNNs) are a subset of Deep Learning, a branch of machine learning that use fully linked layers to achieve the highest accuracy in tasks like object detection and speech recognition.

A CNN model involves two types of layers

1. Convolution layers

2. Fully connected layers.

2.2 **Forward Propagation in Neural Network**

After the input layer, the convolution layer is added. The convolution layer's output is delivered into a fully connected layer network. The convolution layer, which can be thought of as a feature extractor, will perform convolution as well as other operations on the input data, while the fully connected layer can be thought of as a decision block that determines if the input belongs to one of two classes.

2.3 **Convolution Layer in CNN**

CNN's Convolution Layer The kernel is slid over the input signal to perform convolution.

This can be accomplished in one of two ways:

1. Cross-correlation - non-causal convolution (mostly in CNNs)

2. Causal convolution.

2.3.1 Cross correlation- Non causal convolution

Cross-correlation is the same as non-causal convolution. Because the outcome y is contingent on the future input, it is non-causal. Non causal systems, for example, are those in which the output y (0) of a system is contingent on a future input x (1).

Let x represent the input to the convolution layer of length n, and h represent the kernel of length k.

After each convolution operation, change the kernel window s places (number of strides). Then, given stride s, non-causal convolution between x and h is defined as

$$y(n) = \begin{cases} \sum_{i=0}^{k} x(n+i)h(i), & \text{if } n = 0. \\ \sum_{i=0}^{k} x(n+i+(s-1))h(i), & \text{otherwise.} \end{cases}$$

-------------(2.1)

The output length is not equal to the length of the input data, and this state is referred to be "valid." In some circumstances, we wish to use padding to make the output length equal to the input length, and this state is known as "same" padding convolution.

2.3.2 Causal Convolution

In case of causal convolution, the output is not dependent on future inputs.

So let the input to convolution layer of length n be defined by x and let the kernel of length k be defined by h. Let the kernel window be shifted s positions after each convolution operation.

Then causal convolution between x and h for stride s is defined as

$$y(n) = \begin{cases} \sum_{i=0}^{k} x(n-i)h(i), & \text{if } n = k-1. \\ \sum_{i=0}^{k} x(n-i+(s-1))h(i), & \text{otherwise.} \end{cases}$$

------(2.2)

Pooling in CNN

Pooling aids in reducing the dimensionality of a mapping while highlighting the most important elements. To lower the dimension of the convolution output, pooling is usually used after the convolution layer. It also aids in the reduction of overfitting. The max pooling approach is the most popular pooling approach.

After each pooling operation, max pooling will pick up the maximum value in a window of size f, which will be slid over the input with a stride of length s.

2.4 fully connected layer

A fully linked layer is one in which all of the neurons in one layer are connected to all of the neurons in the following layer. The fully connected layer takes the flattened output of the convolution layers and maps it to the output.

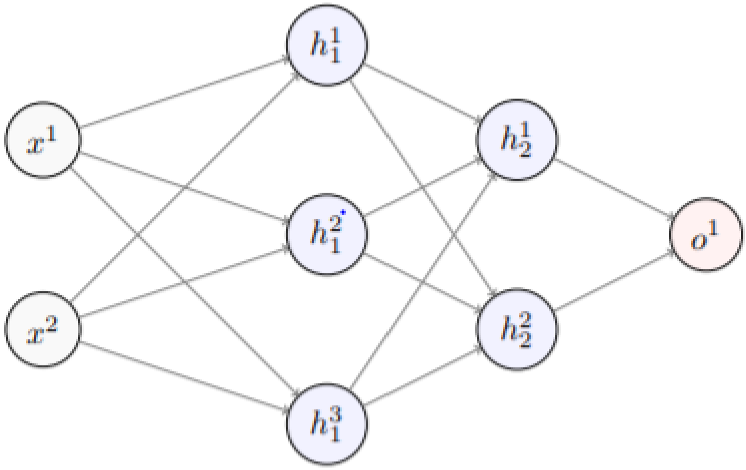


Fig 2.1 Fully connected network

2.5 Different Activation Functions

There are two types of activation functions: linear and non-linear. The inner product of the neuron's input and its weight set is known as the net. The neuron's output is a net activation function indicated by y = f. (net). For more complicated mappings In the network, non-linear activations will be used. There are numerous non-linear functions from which to choose, and the decision will be decided while creating a CNN.

ReLU is a popular and often used activation function (Rectified Linear Unit). Except for the last layer, ReLU is considered to be the activation function for all neurons.

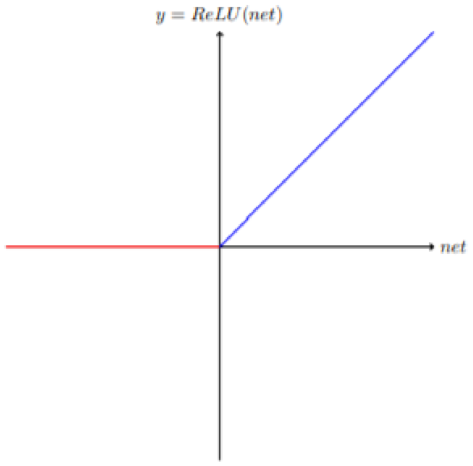


Fig 2.2 ReLU activation function

y = max(0, net) ---------------------------------(2.3)

Prior to the popularity of the ReLU activation function, sigmoid activation (net) was well-known. However, it has been discovered that ReLU computes quicker than sigmoid activations..

Sigmoid activation is given by the following equation.

$$\sigma(net) = \frac{1}{1 + e^{-net}}$$

--------------------- (2.4)

The kind of output determines which activation function is used in the output layer. SoftMax activations are recommended for classification difficulties, while ReLU activations are favoured for predictive/regression problems.

The SoftMax function is used to solve n-class problems.

$$P(class(\underline{i}) = j|\underline{net}) = \frac{e^{net_j}}{\sum_{k=1}^{n} e^{net_k}}$$

---------(2.5)

2.6 Dropouts

Finding a CNN model without dropout regularisation is really tough. Dropouts are employed to prevent overfitting, which is defined as the phenomenon of memorising inputs rather than learning their properties.

The output of a neuron that sends 0 input to the following layer is called dropout. We can determine whether a neuron drops out of the output using the dropout P(drop) rate. If P(drop) = 0.6, each output chooses a number between 0 and 1 at random. If the chosen value is less than 0.6, the output will be discarded.

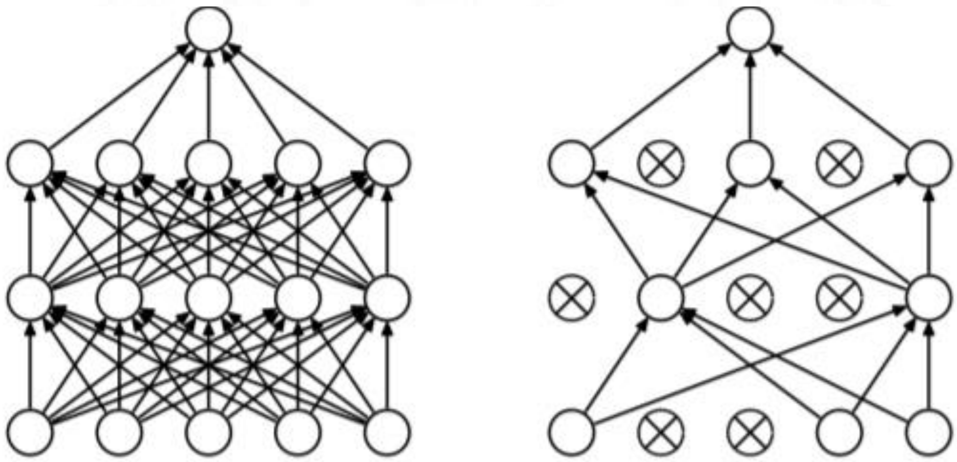


Fig 2.3 Complete neural network After applying dropout

2.7 Back Propagation in Neural Network

2.7.1 Loss Function

The loss function is defined as the difference between the expected output (o) and the actual output (t) (t). Total sum of squares, Mean Squared Error, and Root Mean Square Error are some of the most frequent loss functions.

Expression for Total Sum of Squares

$$L = \frac{1}{2}(\underline{t} - \underline{o})^T(\underline{t} - \underline{o})$$
$$= \frac{1}{2}\sum(t - o)^2$$

---------------------**(2.6)**

MSE and RMSE are loss functions obtained by modifying the preceding equation, and they are utilised for regression-based applications.

Categorical cross entropies, also known as Maximum Likelihood Estimation, are used to solve classification problems.

$$L = -\frac{1}{N}\sum_{n=1}^{N}[y_n log(\hat{y}_n) + (1 - y_n)log(1 - \hat{y}_n)]$$

------------**(2.7)**

Our aim is to get the minimized loss function, for that will use Gradient descent optimization.

2.7.2 Gradient Descent (GDR) Optimization

Loss functions are generally convex functions, and if weights are updated in the opposite direction of gradients, weights achieve global minima after a few epochs. As a result, the gradient of loss with regard to the output should be back propagated to update the weights. The parameter sensitivity j is used to back propagate this loss information from the output layer j to the previous layer i

Classification metrics

Confusion matrix

	Prediction is Positive	Prediction is Negative
Actual Outcome is Positive	True Positive (TP)	False Negative (FN)
Actual Outcome is Negative	False Positive (FP)	True Negative (TN)

Fig 2.4 Confusion Matrix

- **Accuracy:** The ratio between correctly predicted outcomes and the sum of all predictions. Accuracy = **((TP + TN) / (TP + TN + FP + FN))**

- **Precision:** All true positives divided by all positive predictions.

  Precision = (TP / (TP + FP))

- **Recall:** How many positives did the model identify out of all possible positives

  Recall = TP / (TP + FN))

- **F1-score:** This is the weighted average of precision and recall. **(2 x recall x precision / (recall + precision))**

CHAPTER -3

fNIRS-based brain-computer interfaces

3.1 Introduction

A brain-computer interface (BCI) is a communication system that allows users to operate computers or other external devices using their brain activity. A brain-computer interface (BCI) system provides control channels that are independent of the brain's typical output channels. People with motor impairments, such as spinal cord injuries, can use these types of systems to communicate and regain motor capabilities. It will also be utilised as a neuro-rehabilitation system to help such persons improve their physical and cognitive abilities.

There are five stages to a BCI system.

1. Acquisition of brain signals 2. Preprocessing 3. Feature extraction/selection

4. Classification

5. The user interface of the application.

The relevant signals are gathered using a suitable brain-imaging modality during the brain-signal acquisition stage. Because the recorded signals are typically weak and contain sounds (both physiological and instrumental), preprocessing is required to remove the noise and artefacts. Then, in the next stage, some relevant data, referred to as "features," is extracted. In the fourth stage, these characteristics are classified using a suitable classifier. Finally, the classified signals are sent to any external equipment or computers in order to generate the required control orders. A real-time display of brain activity is helpful in neurofeedback applications, allowing self-regulation of brain activities.

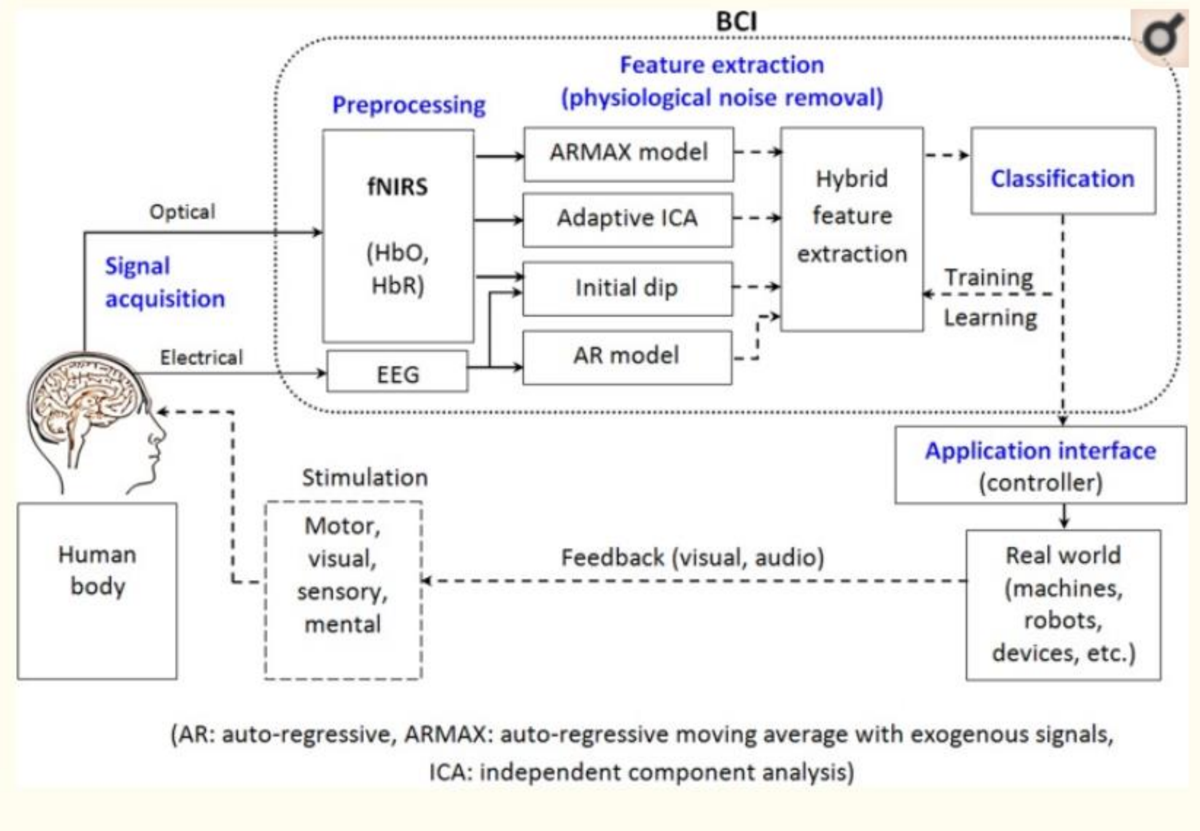


Fig 3.1 **hybrid fNIRS-EEG BCI**

The blood flow variations in the local capillary network generated by neuron firings are calculated using fNIRS. Because haemoglobin is an oxygen carrier, variations in HbR and HbO concentrations following neural activation can be linked to specific neuronal firings. Near-infrared (NI) light emitter-detector pairs of two or more wavelengths are used in fNIRS. Multiple scattering of photons occurs when NI light is released into the scalp and diffuses through the brain tissues. After passing through the cortical region of the brain, where the chromophores (i.e., HbO and HbR) are changing in time, some of these photons depart the head. Using strategically placed detectors, these expelled photons are then recognised. Because HbO and HbR have different absorption coefficients for different wavelengths of NI light, the modified Beer-Lamberts equation can be used to assess changes in HbO and HbR concentrations along the route of photons by connecting the exiting-photon intensity and the incident-photon intensity.

3.2 Brain Signal Acquisition

BCI will capture information about the user's intent using brain signals. Obtaining appropriate brain signals is the first step in constructing a fNIRS-BCI system. The main motor cortex and the prefrontal cortex are the two most common brain sections. The motor cortex receives signals relating to motor execution and motor imagery tasks, whereas the prefrontal cortex receives signals linked to mental counting and music imagery. In these two sectors, a variety of emitter-detector configurations have been employed; however, because the emitter-detector distance is so important in fNIRS measurement, it is normally kept within a certain range. If an increase in the distance between the emitter and the detector equates to an increase in imaging depth. An emitter-detector spacing of roughly 3 cm was preferred to measure hemodynamic response signals from cortical regions. Only skin-layer contribution will be present at a separation of less than 1 cm, however a separation of more than 5 cm may result in weak and useless signals.

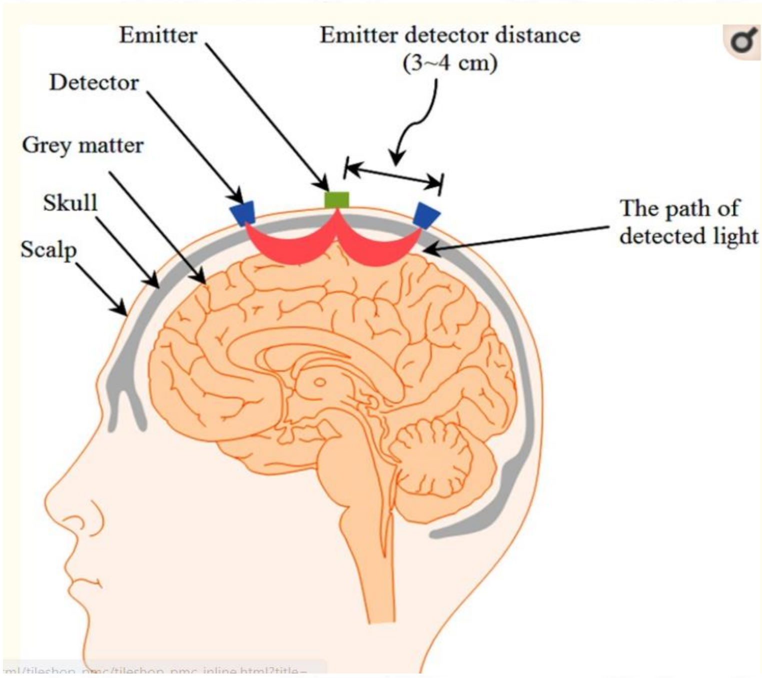


Fig 3.2 emitter-detector pairs showing the banana-shaped paths of light

Motor cortex activities

Because they are natural ways of delivering BCI control over external devices, primary motor cortex activities are the optimum candidate for fNIRS-BCI application. Motor execution and motor imagery are the two most prevalent acquired processes from the motor cortex.

Motor execution

The motor execution job requires producing muscle tensions through muscular motions and entails moving a body part to activate the motor cortex. Because motor execution necessitates muscular contraction.

3.3 Pre-processing

Instrumental noise, experimental error, and physiological noise are three types of noise that can be found in fNIRS results. Because experimental error and instrumental noise are unrelated to brain activity, it is preferable to eliminate them before converting the raw optical density signals to HbO and HbR concentration changes using the modified method. Beer -Lambert law

Removing Instrument noise

Instrumental noise in fNIRS signals is created by the surrounding environment, as well as the system's hardware and instrument degradation. High frequencies are generally involved. A low-pass filter (with a cutoff frequency of 35 Hz, for example) can be used to remove the high frequencies.

Removing experimental errors

Motion artefacts, such as head movements, cause optodes to move from their set places, resulting in experimental mistakes. A spike-like noise can be caused by a quick change in light intensity. Eigenvector-based spatial filtering and wavelet-analysis-based approaches have been presented as solutions for motion-artifact correction.

Physiological noises

Heartbeat (11.5 Hz), breathing (0.20.5 Hz), and Mayer waves (0.1 Hz) are examples of physiological noises that relate to blood pressure changes. To reduce physiological noise, approaches such as band-pass filtering, adaptive filtering, PCA, and independent component analysis (ICA) have been applied.

3.4 Feature extraction/selection

After the data has been pre-processed, the various brain functions are categorised based on particular characteristics. Some features of fNIRS-BCI are taken from detected light-intensity signals, whereas the majority are taken from hemodynamic data. This is because HbO, HbR, total haemoglobin (HbT), and cerebral oxygen exchange (COE = HbO - HbR) all provide more alternatives for selecting appropriate features. It is critical to consider an ideal feature set for classification when performing classification. It is vital to choose qualities that are comparable to a given class while yet being distinct from other classes.

CHAPTER 4

Methodology

4.1 Experimental Procedure

We collected Fnirs data from three healty subjects. For data acquisition, a continuous wave, multichannel fNIRS device with two wavelengths (750nm and 830nm) and a sampling frequency of 7.8125 Hz, was used. Then a bandpass filter with a passbands of 0.01-0.2Hz and 0.01-1Hz was applied to the raw data. We trained our classifiers on fNIRS data for signal classification during epochs of rest, left active, and right active hand movements

The subjects were asked to be seated on a chair facing a computer monitor on which the experimental task was displayed. To generate a robust signal of the neural activity, subjects were requested to perform a motor execution. Subjects were instructed to relax before the paradigm began in order to stabilize blood flow to the cortex, to relax during the rest task and open and close left or right hand depending on the cue presented on the screen.

Each subject underwent a total of 4 sessions.

After pre-proccesing of the data each timeframe consists of total 7 sec data from all the channels. As data collected on 7.81Hz frequency will get about 8 observations for each second. so to label one image we need to consider 56 observations.

4.2 Different types of labelling approches

To make the classifer work in real-time with minimum delay. 3 labelling approaches were used to find the classification accuracy.

1. Labelling approach by considersing latest 1sec data
2. Labelling approach by considersing latest 2sec data
3. Labelling approach by considersing latest 3sec data

Labelling approch by considering latest 1sec data

To label an image with 56 obsevations we look_at latest 8 observations; if all the observations are taken from the same user activity (Example: if all are "Rest" then label it as "Rest") then label it as same belonging to that particular user activity. If all are not same then find which is more frequent in the lastest 3 sec data (24 obesevations) and then label it accordingly (Example: Rest-5, left-12, right-7 then label it as "Left"). If there are 2 classes which are equally frequent and more then label with the latest among those 2 classes (Example: Rest-6, left-9, right-9 then label it as "Left").

Labelling approch by considering latest 2sec data

To label an image with 56 obsevations we look_at latest 16 observations; if all the observations are taken from the same user activity (Example: if all are "Rest" then label it as "Rest") then label it as same belonging to that particular user activity. If all are not same then find which is more frequent in the lastest 3 sec data (24 obesevations) and then label it accordingly (Example: Rest-5, left-12, right-7 then label it as "Left"). If there are 2 classes which are equally frequent and more then label with the latest among those 2 classes (Example: Rest-6, left-9, right-9 then label it as "Left").

Labelling approch by considering latest 3sec data

To label an image with 56 obsevations we look_at latest 24 observations; if all the observations are taken from the same user activity (Example: if all are "Rest" then label it as "Rest") then label it as same belonging to that particular user activity. If all are not same then find which is more frequent in the lastest 3 sec data (24 obesevations) and then label it accordingly (Example: Rest-5, left-12, right-7 then label it as "Left"). If there are 2 classes which are equally frequent and more then label with the latest among those 2 classes (Example: Rest-6, left-9, right-9 then label it as "Left").

Testing was performed on all the 4 sessions individually by considering one session for testing and other sessions for training. Like when I am testing session-4 then classifer is trained with sessions 1,2 3.

1-D Convolution for Time Series

Let's consider a time series of length *n* and width *k*. The length defines the number of timesteps, and the width means number of independent variables (In case of fNIRS it is the number of channels) in a multivariate time series.

The width of the convolution kernel is always equal to the width of the time series, but length of the kernel can be varied. So, the kernel moves in one in time axis from the starting of a time series towards its end, performing convolution. It does not move to the right side or to the left side as it does when we applied to images in the 2-D convolution.

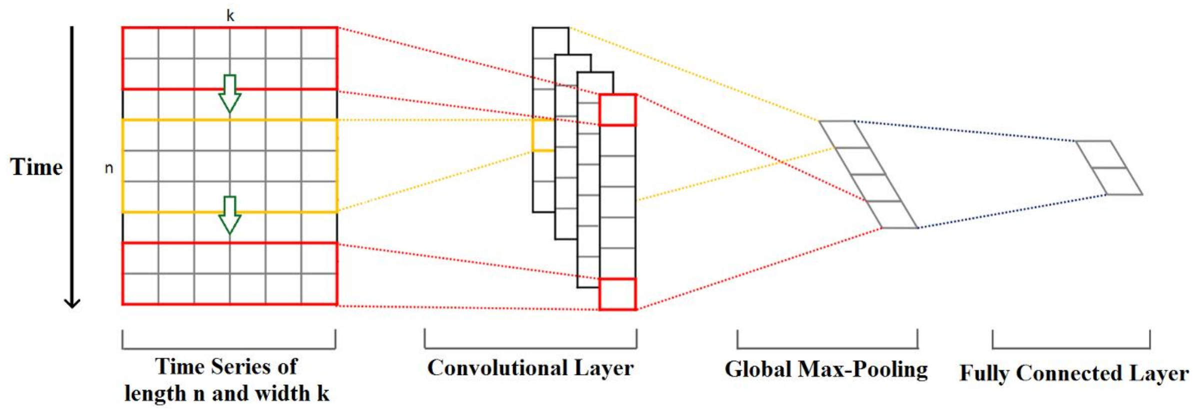


Fig4.1 1-D convolution for Time Series

The values of the kernel will get multiplied by the corresponding weights of the time series. Those multiplications are added together, then apply a nonlinear activation function to the value. The final value will be an element of a new "filtered" univariate time series, and Then kernel will move in the time axis to give the next value. The number of convolution kernels will be same as number of filters we defined in the model. Based on the length of the kernel, various properties, features of the input get captured in each of the new filtered series.

$4.9 = 0.4\times1 + 0.8\times2 + 0.5\times(-3) + 0.7\times1 + 0.7\times(-1) + 0.3\times1 + 0.6\times(-2) + 0.7\times2 + 0.8\times2 + 0.4\times2 + 0.9\times1 + 0.2\times3$

	Feature 1	Feature 2	Feature 3	Feature 4
Timestamp 1	0.4	0.8	0.5	0.7
Timestamp 2	0.7	0.3	0.6	0.7
Timestamp 3	0.8	0.4	0.9	0.2
Timestamp 4	0.1	0.5	0.8	0.5
Timestamp 5	0.4	0.8	0.7	0.1
Timestamp 6	0.3	0.0	0.2	0.6
Timestamp 7	0.2	0.5	0.8	0.2

Filter (kernel size = 3)

1	2	-3	1
-1	1	-2	2
2	2	1	3

Timestamp 1, 2, 3	4.9
Timestamp 2, 3, 4	1.9
Timestamp 3, 4, 5	2.3
Timestamp 4, 5, 6	1.0
Timestamp 5, 6, 7	3.3

	Feature 1	Feature 2	Feature 3	Feature 4
Timestamp 1	0.4	0.8	0.5	0.7
Timestamp 2	0.7	0.3	0.6	0.7
Timestamp 3	0.8	0.4	0.9	0.2
Timestamp 4	0.1	0.5	0.8	0.5
Timestamp 5	0.4	0.8	0.7	0.1
Timestamp 6	0.3	0.0	0.2	0.6
Timestamp 7	0.2	0.5	0.8	0.2

Filter (kernel size = 3)

1	2	-3	1
-1	1	-2	2
2	2	1	3

Timestamp 1, 2, 3	4.9
Timestamp 2, 3, 4	1.9
Timestamp 3, 4, 5	2.3
Timestamp 4, 5, 6	1.0
Timestamp 5, 6, 7	3.3

Fig 4.2 Filter moving in the time axis and performing convolution.

Let's look into each layer.

First 1D CNN layer:

The first layer used a filter of length 3 (also called kernel size). One filter can allow the neural network to learn one single feature in the first layer. That will not be sufficient for getting good results, therefore we use 80 filters. So, the model to train 80 different features on the first layer of the neural network. Each column of the output matrix holds the output of one single filter.

Second 1D CNN layer

The output from the first 1D CNN will be fed into the 2nd CNN layer. In 2nd CNN layer also we define 100 different filters to be trained on this level.

Third 1D CNN layer

The output of the 2nd 1D CNN is given to the 3rd 1D CNN layer in order to learn higher level features. Number of filters used are 80.

Max pooling layer:

Generally, a pooling layer is used after a CNN layer in order to decrease the complexity of the output and prevent overfitting of the data.

Training Accuracy: -

- For all the subject the classifier is giving training accuracy almost more than 99.50%.

Accuracies variation when rerunning the same model multiple times: -

	1st	2nd	3rd	4th	5th
Subject-1	81.78%	80.01%	79.85%	82.11%	81.22%

RESULT

Accuracies for the frequency 0.01-0.2Hz

	Accuracy when consider 1 sec data for labelling	Accuracy when consider 2 sec data for labelling	Accuracy when consider 3 sec data for labelling
Subject 1(training on 1-3 sessions and testing on 4th session)	68.70%	74.71%	67.49%
Subject 1(training on 1&2&4 sessions and testing on 3rd session)	69.59%	68.98%	68.86%
Subject 1(training on 1&3&4 sessions and testing on 2nd session)	56.92%	60.67%	61.86%
Subject 1(training on 2&3&4 sessions and testing on 1st session)	66.57%	69.70%	63.09%
Subject 2(training on 1-3 sessions and testing on 4th session)	61.90%	65.74%	67.82%
Subject 2(training on 1&2&4 sessions and testing on 3rd session)	59.78%	63.32%	65.33%
Subject 2(training on 1&3&4 sessions and testing on 2nd session)	64.45%	61.03%	56.30%
Subject 2(training on 2&3&4 sessions and testing on 1st session)	61.59%	60.68%	63.91%
Subject 3(training on 1-3 sessions and testing on 4th session)	33.26%	35.61%	37.47%

Subject 3(training on 1&2&4 sessions and testing on 3rd session)	40.34%	36.61%	39.91%
Subject 3(training on 1&3&4 sessions and testing on 2nd session)	45.25%	44.39%	44.79%
Subject 3(training on 2&3&4 sessions and testing on 1st session)	35.37%	28.62%	34.09%

Table-4.1 Accuracies of 1CNN layer with all labelling approaches with frequency band (0.01-0.2Hz)

	Accuracy when consider 1 sec data for labelling	Accuracy when consider 2 sec data for labelling	Accuracy when consider 3 sec data for labelling
Subject 1(training on 1-3 sessions and testing on 4th session)	75.23%	78.25%	76.01%
Subject 1(training on 1&2&4 sessions and testing on 3rd session)	75.16%	76.01%	75.71%
Subject 1(training on 1&3&4 sessions and testing on 2nd session)	63.42%	61.98%	61.34%
Subject 1(training on 2&3&4 sessions and testing on 1st session)	68.95%	74.79%	77.38%
Subject 2(training on 1-3 sessions and testing on 4th session)	67.88%	71.66%	70.96%
Subject 2(training on 1&2&4 sessions and testing on 3rd session)	68.78%	72.17%	71.62%
Subject 2(training on 1&3&4 sessions and testing on 2nd session)	64.60%	73.05%	62.44%
Subject 2(training on 2&3&4 sessions and testing on 1st session)	66.08%	59.88%	63.30%
Subject 3(training on 1-3 sessions and testing on 4th session)	37.77%	33.72%	34.45%
Subject 3(training on 1&2&4 sessions and testing on 3rd session)	50.60%	37.19%	39.51%

Subject 3(training on 1&3&4 sessions and testing on 2nd session)	47.01%	42.26%	46.28%
Subject 3(training on 2&3&4 sessions and testing on 1st session)	30.12%	30.00%	30.94%

Table-4.2 Accuracies of 2CNN layers with all labelling approaches with frequency band (0.01-0.2Hz)

	Accuracy when consider 1 sec data for labelling	Accuracy when consider 2 sec data for labelling	Accuracy when consider 3 sec data for labelling
Subject 1(training on 1-3 sessions and testing on 4th session)	78.16%	77.79%	77.67%
Subject 1(training on 1&2&4 sessions and testing on 3rd session)	75.31%	72.63%	80.12%
Subject 1(training on 1&3&4 sessions and testing on 2nd session)	62.41%	61.74%	69.10%
Subject 1(training on 2&3&4 sessions and testing on 1st session)	71.53%	68.39%	69.03%
Subject 2(training on 1-3 sessions and testing on 4th session)	76.37%	67.36%	66.47%
Subject 2(training on 1&2&4 sessions and testing on 3rd session)	65.49%	73.30%	71.41%
Subject 2(training on 1&3&4 sessions and testing on 2nd session)	61.21%	69.30%	65.40%
Subject 2(training on 2&3&4 sessions and testing on 1st session)	63.97%	67.14%	65.16%
Subject 3(training on 1-3 sessions and testing on 4th session)	39.21%	38.54%	31.62%
Subject 3(training on 1&2&4 sessions and testing on 3rd session)	46.26%	36.34%	49.28%

Subject 3(training on 1&3&4 sessions and testing on 2nd session)	45.67%	44.24%	44.91%
Subject 3(training on 2&3&4 sessions and testing on 1st session)	33.90%	24.26%	22.58%

Table-4.3 Accuracies of 3CNN layers with all labelling approaches with frequency band (0.01-0.2Hz)

Data collected with frequency 0.01-1.0Hz

	Accuracy when consider 1 sec data for labelling	Accuracy when consider 2 sec data for labelling	Accuracy when consider 3 sec data for labelling
Subject 1(training on 1-3 sessions and testing on 4th session)	80.03%	81.81%	82.72%
Subject 1(training on 1&2&4 sessions and testing on 3rd session)	75.04%	78.87%	78.72%
Subject 1(training on 1&3&4 sessions and testing on 2nd session)	71.05%	73.89%	76.21%
Subject 1(training on 2&3&4 sessions and testing on 1st session)	74.95%	76.84%	79.18%
Subject 2(training on 1-3 sessions and testing on 4th session)	73.28%	76.27%	75.59%
Subject 2(training on 1&2&4 sessions and testing on 3rd session)	81.35%	82.06%	83.92%
Subject 2(training on 1&3&4 sessions and testing on 2nd session)	68.57%	70.49%	73.45%
Subject 2(training on 2&3&4 sessions and testing on 1st session)	71.78%	74.01%	71.93%
Subject 3(training on 1-3 sessions and testing on 4th session)	58.60%	57.62%	57.01%

Subject 3(training on 1&2&4 sessions and testing on 3rd session)	57.50%	53.93%	53.11%
Subject 3(training on 1&3&4 sessions and testing on 2nd session)	45.31%	44.43%	48.14%
Subject 3(training on 2&3&4 sessions and testing on 1st session)	30.39%	31.74%	24.99%

Table-4.4 Accuracies of 1CNN layer with all labelling approaches with frequency band (0.01-1Hz)

	Accuracy when consider 1 sec data for labelling	Accuracy when consider 2 sec data for labelling	Accuracy when consider 3 sec data for labelling
Subject 1(training on 1-3 sessions and testing on 4th session)	79.85%	84.47%	80.63%
Subject 1(training on 1&2&4 sessions and testing on 3rd session)	75.22%	79.48%	77.90%
Subject 1(training on 1&3&4 sessions and testing on 2nd session)	68.37%	71.18%	73.86%
Subject 1(training on 2&3&4 sessions and testing on 1st session)	69.86%	73.33%	74.31%
Subject 2(training on 1-3 sessions and testing on 4th session)	72.88%	76.97%	78.10%
Subject 2(training on 1&2&4 sessions and testing on 3rd session)	72.66%	78.76%	78.97%
Subject 2(training on 1&3&4 sessions and testing on 2nd session)	71.71%	74.92%	79.07%
Subject 2(training on 2&3&4 sessions and testing on 1st session)	71.32%	70.44%	71.51%
Subject 3(training on 1-3 sessions and testing on 4th session)	57.22%	57.28%	58.50%

Subject 3(training on 1&2&4 sessions and testing on 3rd session)	60.02%	58.93%	61.30%
Subject 3(training on 1&3&4 sessions and testing on 2nd session)	54.02%	56.15%	53.71%
Subject 3(training on 2&3&4 sessions and testing on 1st session)	25.33%	26.24%	27.77%

Table-4.5 Accuracies of 2CNN layers with all labelling approaches with frequency band (0.01-1Hz)

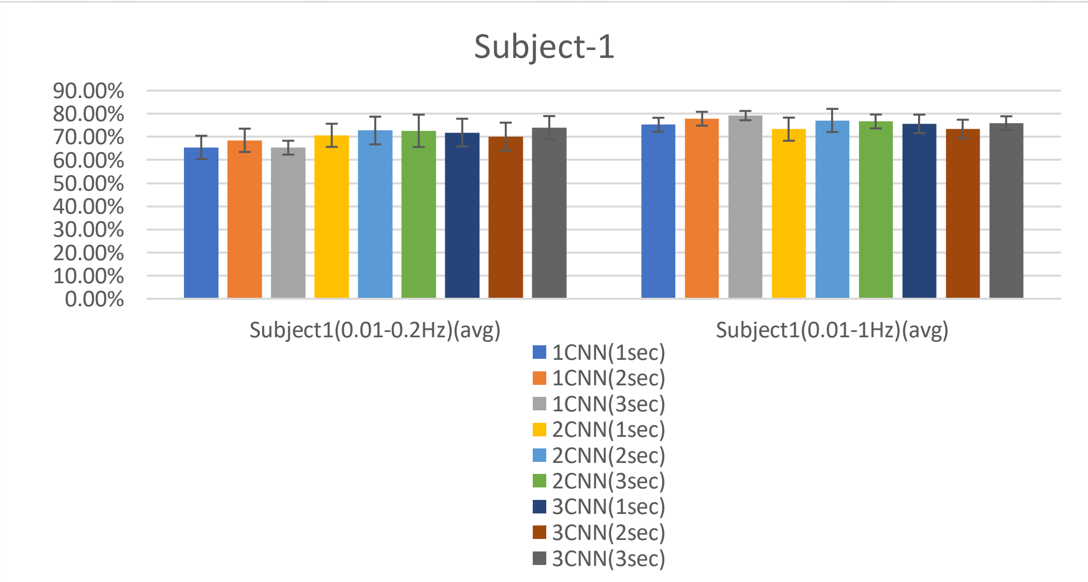
	Accuracy when consider 1 sec data for labelling	Accuracy when consider 2 sec data for labelling	Accuracy when consider 3 sec data for labelling
Subject 1(training on 1-3 sessions and testing on 4th session)	79.09%	78.43%	80.09%
Subject 1(training on 1&2&4 sessions and testing on 3rd session)	77.99%	75.53%	75.80%
Subject 1(training on 1&3&4 sessions and testing on 2nd session)	69.19%	67.63%	74.63%
Subject 1(training on 2&3&4 sessions and testing on 1st session)	76.20%	72.11%	73.13%
Subject 2(training on 1-3 sessions and testing on 4th session)	77.43%	77.49%	77.09%
Subject 2(training on 1&2&4 sessions and testing on 3rd session)	73.05%	75.28%	78.85%
Subject 2(training on 1&3&4 sessions and testing on 2nd session)	72.47%	73.94%	66.86%
Subject 2(training on 2&3&4 sessions and testing on 1st session)	68.64%	72.79%	73.58%

Subject 3(training on 1-3 sessions and testing on 4th session)	54.05%	53.47%	50.50%
Subject 3(training on 1&2&4 sessions and testing on 3rd session)	62.74%	64.35%	53.75%
Subject 3(training on 1&3&4 sessions and testing on 2nd session)	55.23%	56.18%	52.74%
Subject 3(training on 2&3&4 sessions and testing on 1st session)	31.03%	28.38%	26.98%

Table-4.6 Accuracies of 3CNN layers with all labelling approaches with frequency band (0.01-1Hz)

Graphical repersentation of result:

Fig 4.3 Subject -1 avgerage accuracies with standard deviation error bars



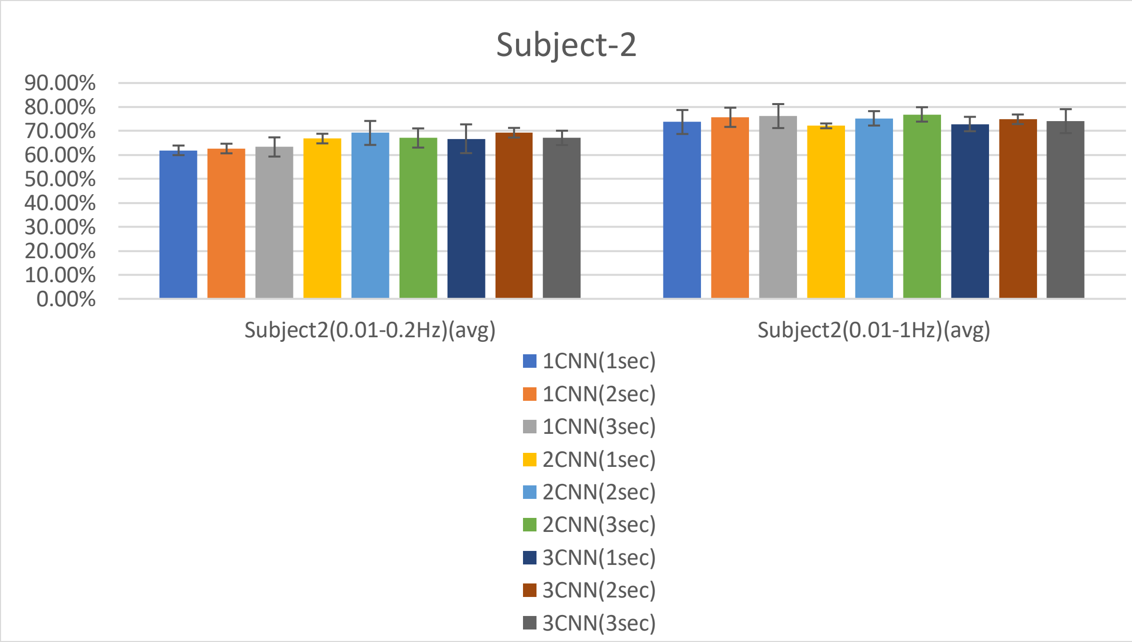


Fig 4.4 Subject -2 avgerage accuracies with standard deviation error bar

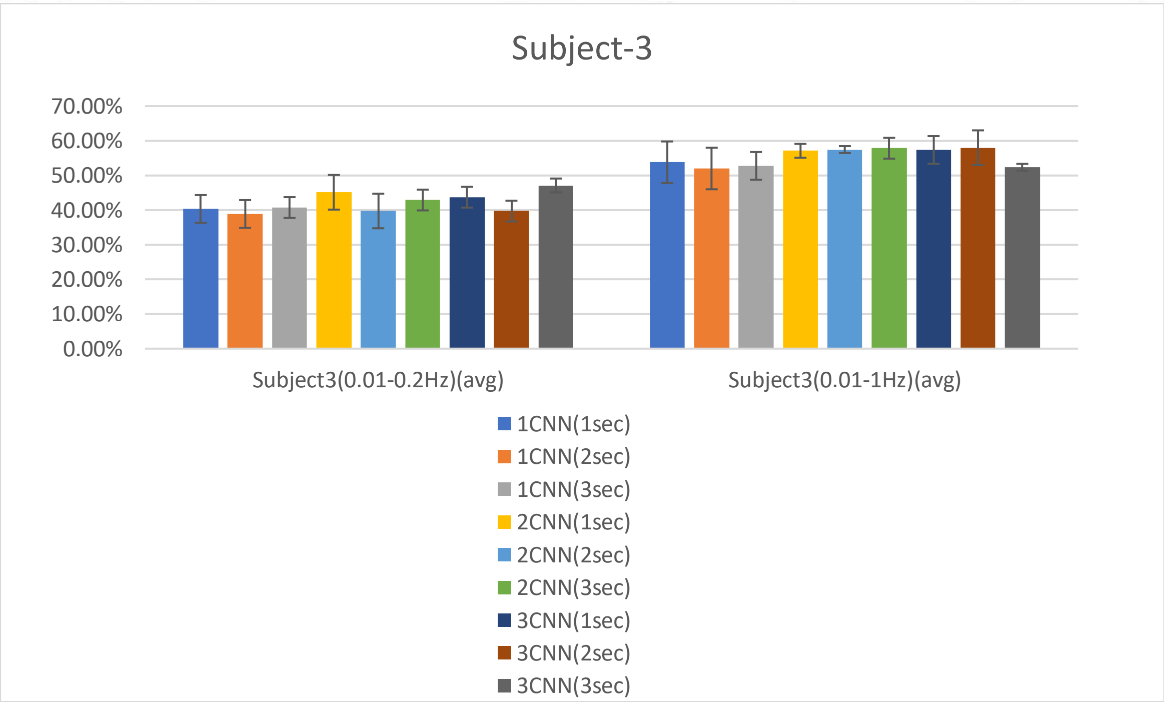


Fig 4.5 Subject -3 average accuracies with standard deviation error bars

CHAPTER-5

Statistical Analysis:

5.1. Subject-1 and Subject-2 combination

Univariate Analysis of Variance

Between-Subjects Factors

		N
Labelling	1	48
	2	48
	3	48
CNN	1	48
	2	48
	3	48
Freq_band	1	72
	2	72

Descriptive Statistics

Dependent Variable: Accuracy

Labelling	CNN	Freq_band	Mean	Std. Deviation	N
1	1	1	63.6875	4.43230	8
		2	74.5063	4.37446	8
		Total	69.0969	7.02211	16
	2	1	68.7625	4.41807	8
		2	72.7338	3.52940	8
		Total	70.7481	4.37352	16
	3	1	69.3063	6.81828	8
		2	74.2575	4.01803	8
		Total	71.7819	5.98049	16
	Total	1	67.2521	5.72270	24
		2	73.8325	3.89478	24
		Total	70.5423	5.87412	48
2	1	1	65.6038	5.15254	8
		2	76.7800	4.02493	8
		Total	71.1919	7.29783	16
	2	1	70.9738	6.57965	8
		2	76.1938	4.69880	8
		Total	73.5838	6.14593	16
	3	1	69.7063	4.84414	8
		2	74.1500	3.40632	8
		Total	71.9281	4.65094	16
	Total	1	68.7613	5.82053	24
		2	75.7079	4.06294	24
		Total	72.2346	6.08093	48
3	1	1	64.3325	4.07049	8
		2	77.7150	4.22823	8
		Total	71.0237	7.98955	16
	2	1	69.8450	6.58550	8
		2	76.7938	3.16810	8
		Total	73.3194	6.14806	16
	3	1	70.5450	5.60169	8
		2	75.0038	4.09962	8
		Total	72.7744	5.27145	16
	Total	1	68.2408	5.98722	24

		2	76.5042	3.86434	24
		Total	72.3725	6.50259	48
Total	1	1	64.5413	4.44557	24
		2	76.3337	4.25324	24
		Total	70.4375	7.35046	48
	2	1	69.8604	5.75901	24
		2	75.2404	4.11173	24
		Total	72.5504	5.64744	48
	3	1	69.8525	5.57841	24
		2	74.4704	3.70276	24
		Total	72.1615	5.23282	48
	Total	1	68.0847	5.79597	72
		2	75.3482	4.04611	72
		Total	71.7165	6.17164	144

Tests of Between-Subjects Effects

Dependent Variable: Accuracy

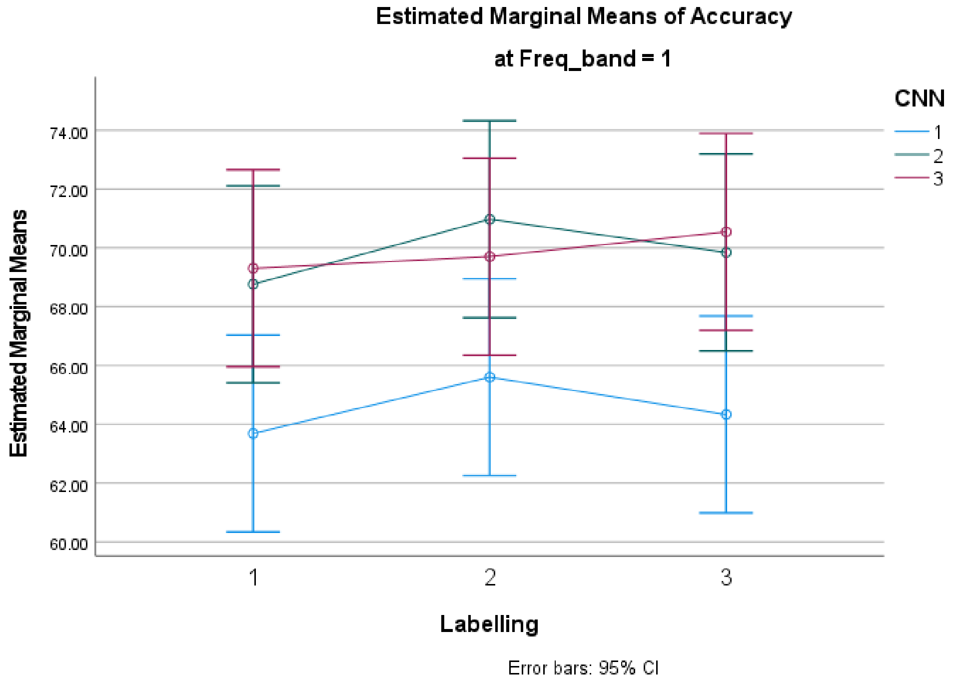
Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	2558.436[a]	17	150.496	6.565	<.001	.470
Intercept	740628.057	1	740628.057	32309.157	<.001	.996
Labelling	99.721	2	49.860	2.175	.118	.033
CNN	121.404	2	60.702	2.648	.075	.040
Freq_band	1899.289	1	1899.289	82.855	<.001	.397
Labelling * CNN	31.345	4	7.836	.342	.849	.011
Labelling * Freq_band	18.800	2	9.400	.410	.664	.006
CNN * Freq_band	372.702	2	186.351	8.129	<.001	.114
Labelling * CNN * Freq_band	15.176	4	3.794	.166	.956	.005
Error	2888.318	126	22.923			
Total	746074.812	144				
Corrected Total	5446.755	143				

a. R Squared = .470 (Adjusted R Squared = .398)

Table 5.1 3-way repeated ANOVA test report for Subject-1 and Subject-2 combined

Profile Plots

Labelling * CNN * Freq_band



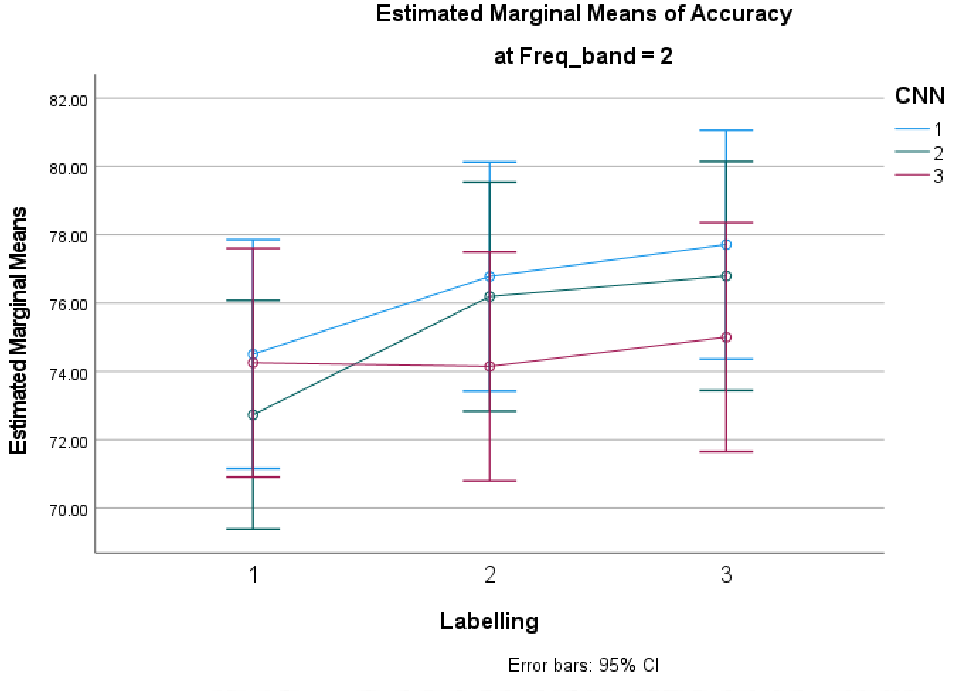


Fig5.1 Interaction plot of 3way repeated ANOVA for subject-1 and subject-2 combined

5.2 Subject-1

Univariate Analysis of Variance

Between-Subjects Factors

		N
Labelling	1	24
	2	24
	3	24
CNN	1	24
	2	24
	3	24
Freq_band	1	36
	2	36

Descriptive Statistics

Dependent Variable: Accuracy

Labelling	CNN	Freq_band	Mean	Std. Deviation	N
1	1	1	65.4450	5.82287	4
		2	75.2675	3.67973	4
		Total	70.3563	6.92100	8
	2	1	70.6900	5.67077	4
		2	73.3250	5.25119	4
		Total	72.0075	5.25200	8
	3	1	71.8525	6.85574	4
		2	75.6175	4.44744	4
		Total	73.7350	5.71581	8
	Total	1	69.3292	6.26982	12
		2	74.7367	4.20916	12
		Total	72.0329	5.90781	24
2	1	1	68.5150	5.81786	4
		2	77.8525	3.33785	4
		Total	73.1838	6.64771	8
	2	1	72.7575	7.32648	4
		2	77.1150	6.03434	4
		Total	74.9363	6.63591	8
	3	1	70.1375	6.79080	4
		2	73.4250	4.64730	4
		Total	71.7813	5.66635	8

	Total	1	70.4700	6.30714	12
		2	76.1308	4.79083	12
		Total	73.3004	6.19370	24
3	1	1	65.3250	3.37564	4
		2	79.2075	2.68086	4
		Total	72.2662	7.93900	8
	2	1	72.6100	7.54842	4
		2	76.6750	3.19688	4
		Total	74.6425	5.78970	8
	3	1	73.9800	5.76289	4
		2	75.9125	2.99173	4
		Total	74.9462	4.37449	8
	Total	1	70.6383	6.59129	12
		2	77.2650	3.05815	12
		Total	73.9517	6.05858	24
Total	1	1	66.4283	4.89525	12
		2	77.4425	3.40670	12
		Total	71.9354	6.97552	24
	2	1	72.0192	6.31797	12
		2	75.7050	4.83358	12
		Total	73.8621	5.81451	24
	3	1	71.9900	6.09489	12
		2	74.9850	3.88187	12
		Total	73.4875	5.22620	24
	Total	1	70.1458	6.23371	36
		2	76.0442	4.10131	36
		Total	73.0950	6.02228	72

Tests of Between-Subjects Effects

Dependent Variable: Accuracy

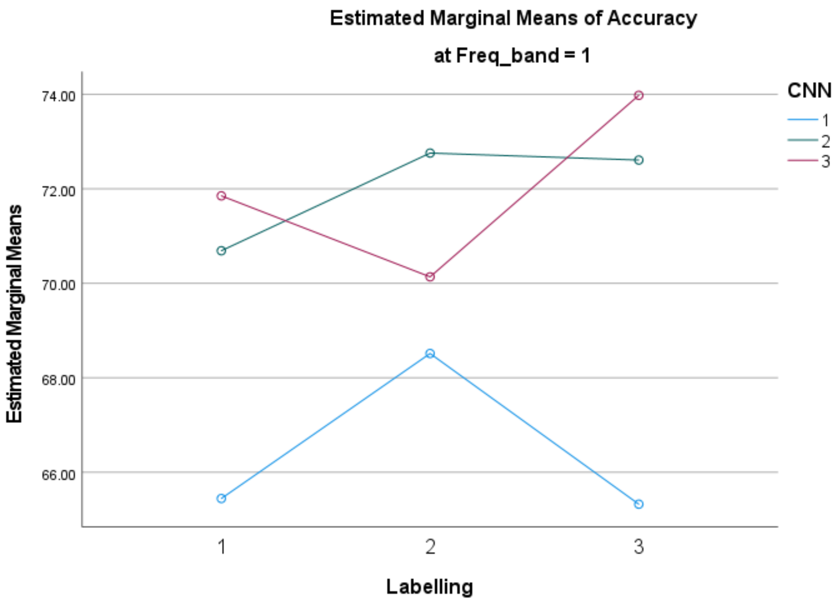
Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	1060.940[a]	17	62.408	2.226	.013	.412
Intercept	384687.290	1	384687.290	13719.957	<.001	.996
Labelling	45.698	2	22.849	.815	.448	.029
CNN	50.091	2	25.045	.893	.415	.032
Freq_band	626.226	1	626.226	22.334	<.001	.293
Labelling * CNN	70.017	4	17.504	.624	.647	.044
Labelling * Freq_band	4.967	2	2.483	.089	.915	.003
CNN * Freq_band	236.978	2	118.489	4.226	.020	.135
Labelling * CNN * Freq_band	26.964	4	6.741	.240	.914	.017
Error	1514.080	54	28.039			
Total	387262.310	72				
Corrected Total	2575.020	71				

a. R Squared = .412 (Adjusted R Squared = .227)

Fig 5.2 3-way repeated ANOVA test report for Subject-1

Profile Plots

Labelling * CNN * Freq_band



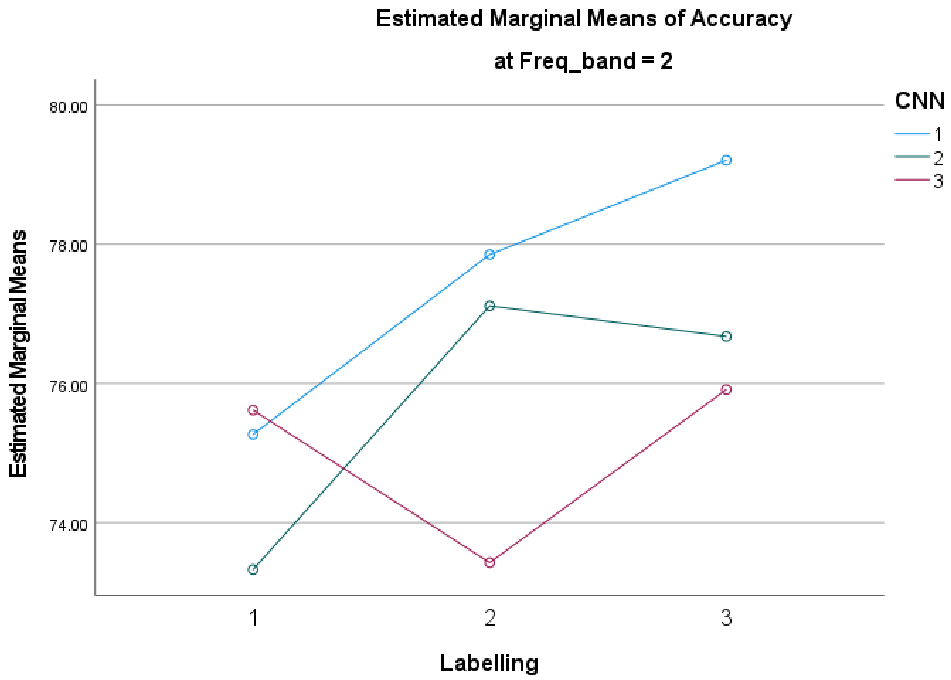


Fig5.2 Interaction plot of 3way repeated ANOVA for subject-1

Subject-2

Univariate Analysis of Variance

Between-Subjects Factors

		N
Labelling	1	24
	2	24
	3	24
CNN	1	24
	2	24
	3	24
Freq_band	1	36
	2	36

Descriptive Statistics

Dependent Variable: Accuracy

Labelling	CNN	Freq_band	Mean	Std. Deviation	N
1	1	1	61.9300	1.92262	4
		2	73.7450	5.43734	4
		Total	67.8375	7.35791	8
	2	1	66.8350	1.86550	4
		2	72.1425	.74728	4
		Total	69.4888	3.12718	8
	3	1	66.7600	6.64709	4
		2	72.8975	3.59981	4
		Total	69.8288	5.93735	8
	Total	1	65.1750	4.44433	12
		2	72.9283	3.49528	12
		Total	69.0517	5.56521	24
2	1	1	62.6925	2.34486	4
		2	75.7075	4.85711	4
		Total	69.2000	7.80156	8
	2	1	69.1900	6.23316	4
		2	75.2725	3.58336	4
		Total	72.2313	5.72054	8
	3	1	69.2750	2.85346	4
		2	74.8750	2.01856	4
		Total	72.0750	3.76773	8
	Total	1	67.0525	4.96854	12
		2	75.2850	3.34265	12
		Total	71.1687	5.90174	24
3	1	1	63.3400	4.96377	4
		2	76.2225	5.34677	4
		Total	69.7813	8.38024	8
	2	1	67.0800	4.88139	4
		2	76.9125	3.62791	4
		Total	71.9962	6.59356	8
	3	1	67.1100	2.92268	4
		2	74.0950	5.29750	4
		Total	70.6025	5.44318	8
	Total	1	65.8433	4.35503	12
		2	75.7433	4.53966	12
		Total	70.7933	6.67045	24
Total	1	1	62.6542	3.09672	12
		2	75.2250	4.85148	12

	Total	68.9396	7.55429	24
2	1	67.7017	4.38895	12
	2	74.7758	3.39341	12
	Total	71.2388	5.27019	24
3	1	67.7150	4.23676	12
	2	73.9558	3.60842	12
	Total	70.8354	4.99724	24
Total	1	66.0236	4.53301	36
	2	74.6522	3.92334	36
	Total	70.3379	6.04919	72

Tests of Between-Subjects Effects

Dependent Variable: Accuracy

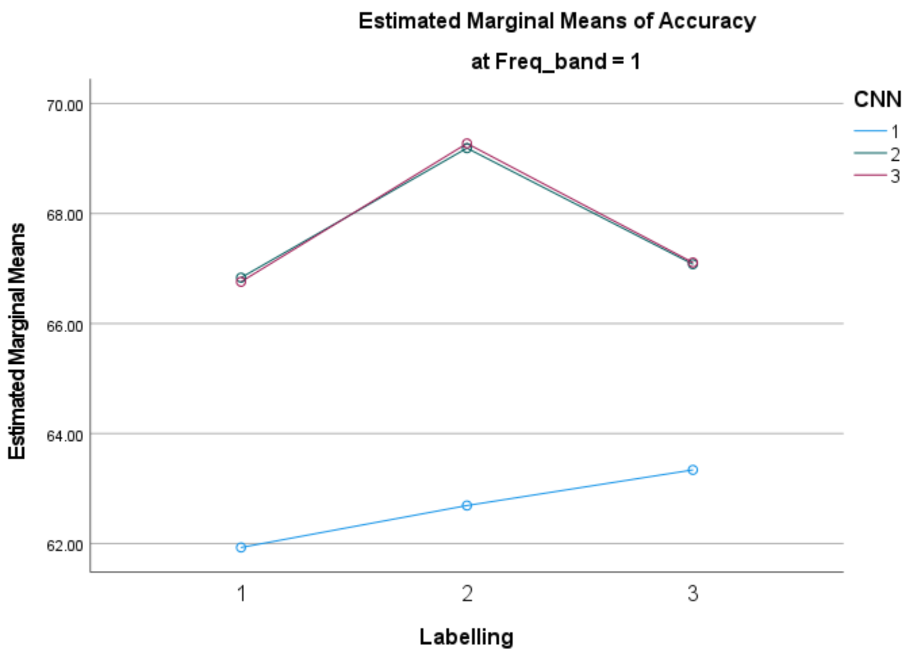
Source	Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	1655.289[a]	17	97.370	5.577	<.001	.637
Intercept	356214.422	1	356214.422	20402.785	<.001	.997
Labelling	61.251	2	30.626	1.754	.183	.061
CNN	72.344	2	36.172	2.072	.136	.071
Freq_band	1340.153	1	1340.153	76.760	<.001	.587
Labelling * CNN	12.480	4	3.120	.179	.948	.013
Labelling * Freq_band	15.237	2	7.618	.436	.649	.016
CNN * Freq_band	141.953	2	70.977	4.065	.023	.131
Labelling * CNN * Freq_band	11.871	4	2.968	.170	.953	.012
Error	942.792	54	17.459			
Total	358812.502	72				
Corrected Total	2598.081	71				

a. R Squared = .637 (Adjusted R Squared = .523)

Fig 5.3 3-way repeated ANOVA test report for Subject-2

Interaction Plots

Labelling * CNN * Freq_band



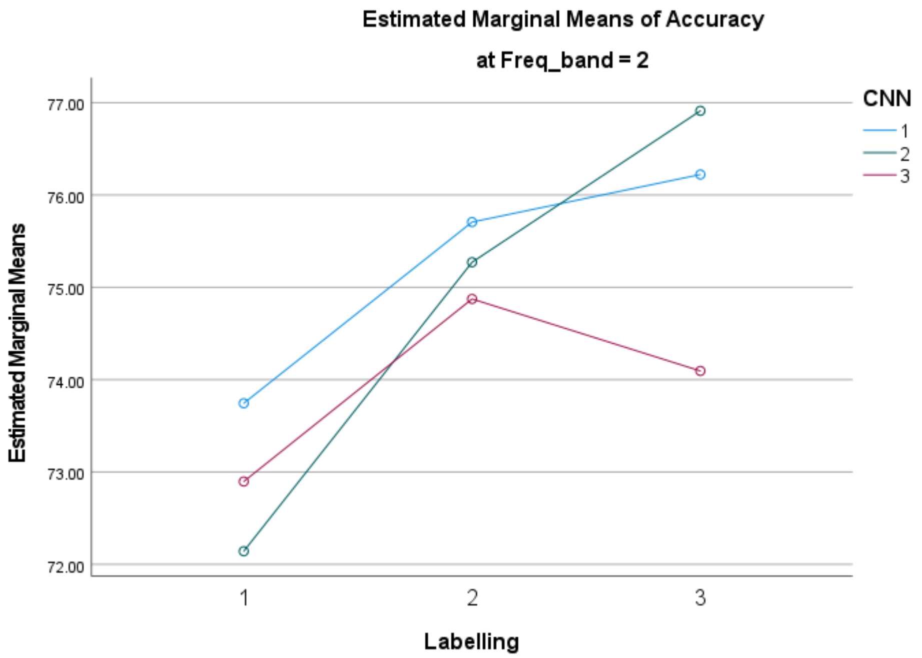


Fig5.3 Interaction plots of 3way repeated ANOVA for subject-2

Subjecct-3

Univariate Analysis of Variance

Between-Subjects Factors

		N
Labelling	1	18
	2	17
	3	15
CNN	1	18
	2	16
	3	16
Freq_band	1	24
	2	26

Descriptive Statistics

Dependent Variable: Accuracy

Labelling	CNN	Freq_band	Mean	Std. Deviation	N
1	1	1	40.3200	4.94003	3
		2	53.4767	7.07280	3
		Total	46.8983	9.03884	6
	2	1	45.1267	6.61910	3
		2	57.1667	3.09156	3
		Total	51.1467	8.05211	6
	3	1	43.5800	3.81677	3
		2	57.1467	4.92325	3
		Total	50.3633	8.41064	6
	Total	1	43.0089	5.02110	9
		2	55.9300	4.93364	9
		Total	49.4694	8.21661	18
2	1	1	38.8700	4.80654	3
		2	51.7900	6.55735	3
		Total	45.3300	8.74749	6
	2	1	39.7250	3.58503	2
		2	57.8600	1.49643	3
		Total	50.6060	10.14871	5
	3	1	39.7067	4.07717	3
		2	57.0100	6.96221	3

		Total	48.3583	10.76382	6
	Total	1	39.3975	3.65749	8
		2	55.5533	5.61512	9
		Total	47.9506	9.52410	17
3	1	1	40.7233	3.72716	3
		2	52.8233	4.54678	3
		Total	46.7733	7.59928	6
	2	1	42.8950	4.78711	2
		2	56.3533	4.28731	3
		Total	50.9700	8.32212	5
	3	1	53.9400	6.59024	2
		2	53.2450	.71418	2
		Total	53.5925	3.84813	4
	Total	1	45.1200	7.27564	7
		2	54.2525	3.77996	8
		Total	49.9907	7.21603	15
Total	1	1	39.9711	4.00776	9
		2	52.6967	5.38206	9
		Total	46.3339	8.00352	18
	2	1	42.9457	5.13833	7
		2	57.1267	2.82330	9
		Total	50.9225	8.22197	16
	3	1	44.7175	7.12269	8
		2	56.1200	4.89900	8
		Total	50.4187	8.33946	16
	Total	1	42.4208	5.67288	24
		2	55.2835	4.73244	26
		Total	49.1094	8.28677	50

Tests of Between-Subjects Effects

Dependent Variable: Accuracy

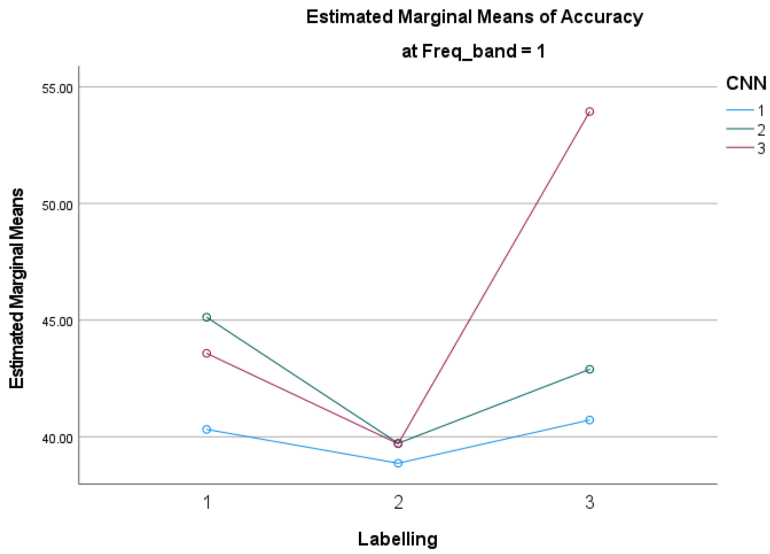
Source	Type III Sum of Squares	Df	Mean Square	F	Sig.	Partial Eta Squared
Corrected Model	2579.174[a]	17	151.716	6.179	<.001	.767
Intercept	116624.664	1	116624.664	4749.985	<.001	.993
Labelling	55.874	2	27.937	1.138	.333	.066
CNN	186.478	2	93.239	3.798	.033	.192
Freq_band	1881.096	1	1881.096	76.615	<.001	.705
Labelling * CNN	37.486	4	9.372	.382	.820	.046
Labelling * Freq_band	118.835	2	59.417	2.420	.105	.131
CNN * Freq_band	39.324	2	19.662	.801	.458	.048
Labelling * CNN * Freq_band	133.718	4	33.430	1.362	.269	.145
Error	785.684	32	24.553			
Total	123951.517	50				
Corrected Total	3364.859	49				

a. R Squared = .767 (Adjusted R Squared = .642)

Fig 5.2 3-way repeated ANOVA test report for Subject-3

Interaction Plots

Labelling * CNN * Freq_band



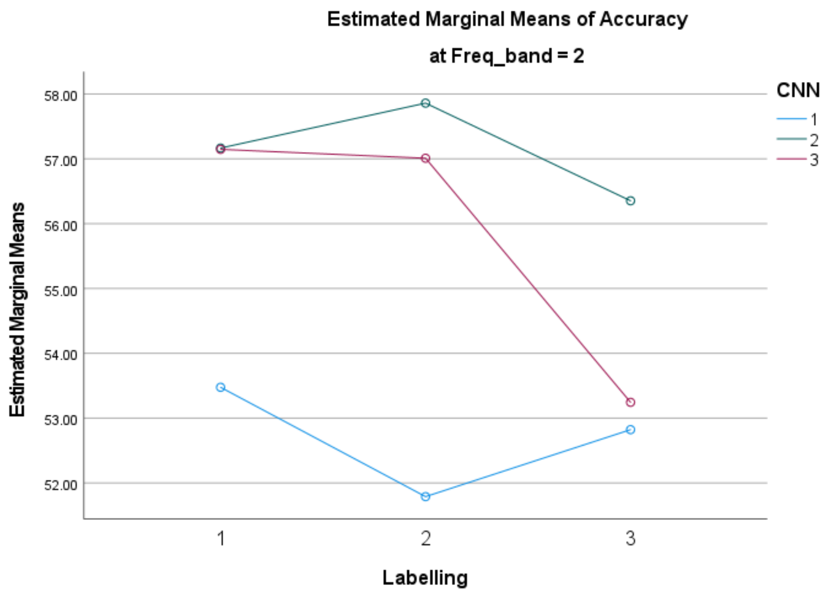


Fig5.4 Interaction plot of 3way repeated ANOVA for subject-3

DISCUSSION:

Subject-1 and Subject-2 combined:

After the doing the stastatical analaysis on the result will get the interaction plots for labelling approach, CNN layers and frequency band. From the interaction plots I found that the optimal labelling approach based on the latest 2 seconds of data in the band 0.01-1Hz and 1 layers of CNN gave the best overall performance.

Subject-1

From the interaction plots i found that the optimal labelling approach based on the latest 3 seconds of data in the band 0.01-1Hz and 1-layers of CNN gave the best overall performance.

Subject-2

From the interaction plots i found that the optimal labelling approach based on the latest 3 seconds of data in the band 0.01-1Hz and 2-layers of CNN gave the best overall performance.

Subject-3

From the interaction plots i found that the optimal labelling approach based on the latest 2 seconds of data in the band 0.01-1Hz and 2-layers of CNN gave the best overall performance.

Conclusion:

From this study I am concluding that the Optimum labelling approach, number of CNN layers and band of frequency is varying from the subject to subject as per the data.

Subject -1:

Optimal labelling approach – 3seconds latest data

Number of CNN layers - 1CNN

Pass band frequency - 0.01 -1Hz

Subject-2:

Optimal labelling approach – 3seconds latest data

Number of CNN layers - 2CNN

Pass band frequency - 0.01 -1Hz

Subject-3:

Optimal labelling approach – 2seconds latest data

Number of CNN layers - 2CNN

Pass band frequency - 0.01 -1Hz

Subject-1 and Subject-2 combination:

Optimal labelling approach – 2seconds latest data

Number of CNN layers - 1CNN

Pass band frequency - 0.01 -1Hz

To generalize the model for any subject optimal labelling approach based on the latest 2 seconds of data in the band 0.01-1Hz and 1-layer (or) 2CNN layer of CNN gave the best overall performance. With lower data as for Subject 3 CNN2 and Label 2 gave best results. Overall, it appears if there is channel loss this configuration is more robust.

REFERENCES:

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, ‘Braincomputer interfaces for communication and control’, Clin Neurophysiol, vol. 113, no. 6, pp. 767–791, Jun. 2002, doi: 10.1016/s1388- 2457(02)00057-3.

[2] E. C. Leuthardt, G. Schalk, J. R. Wolpaw, J. G. Ojemann, and D. W. Moran, ‘A brain–computer interface using electrocorticographic signals in humans’, J. Neural Eng., vol. 1, no. 2, pp. 63–71, Jun. 2004, doi: 10.1088/1741-2560/1/2/001.

[3] C. Kesavadas et al., ‘Real-time functional MR imaging (fMRI) for presurgical evaluation of paediatric epilepsy’, Pediatr Radiol, vol. 37, no. 10, pp. 964–974, Oct. 2007, doi: 10.1007/s00247- 007-0556-4.

[4] N. Naseer and K.-S. Hong, ‘fNIRS-based brain computer interfaces: a review’, Front Hum Neurosci, vol. 9, Jan. 2015, doi: 10.3389/fnhum.2015.00003.

[5] J. Mellinger et al., ‘An MEG-based brain– computer interface (BCI)’, NeuroImage, vol. 36, no. 3, pp. 581–593, Jul. 2007, doi: 10.1016/j.neuroimage.2007.03.019

[6] R. Sitaram et al., ‘Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain–computer interface’, NeuroImage, vol. 34, no. 4, pp. 1416– 1427, Feb. 2007, doi:

[7] Janson Brownlee, ‘1D Convolutional Neural Network Models for Human Activity Recognition’, September 21, 2018, in deep learning for Time series:

[8] N. Naseer, F. M. Noori, N. K. Qureshi, and K.-S. Hong, ‘Determining Optimal Feature Combination for LDA Classification of Functional Near-Infrared Spectroscopy Signals in Brain Computer Interface Application’, Front. Hum. Neurosci., vol. 10, 2016, doi: 10.3389/fnhum.2016.00237

[9] F. Irani, S. M. Platek, S. Bunce, A. C. Ruocco, and D. Chute, ‘Functional near infrared spectroscopy (fNIRS): an emerging neuroimaging technology with important applications for the study of brain disorders’, Clin Neuropsychol, vol. 21, no. 1, pp. 9–37, Jan. 2007, doi: 10.1080/13854040600910018

[10] Convolutional Neural Networks for Visual Recognition. Cs231n: Convolutional neural networks for visual recognition, 2018. [Online; accessed 17-June-2018].

[11] Deep learning. Deep learning — Wikipedia, the free encyclopedia, 2018. [Online; accessed 17-June-2018].

[12] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning, 2016. cite arxiv:1603.07285.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, Nov 1998.

[16] Henry W. Lin and Max Tegmark. Why does deep and cheap learning work so well? CoRR, abs/1608.08225, 2016.

[17] Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, and Jun Zhang. Implementation of training convolutional neural networks. CoRR, abs/1506.01195, 2015.

[18] Richard G. Lyons. Understanding Digital Signal Processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1996.