Enhancing the Decoding Performance of Steady-State Visual Evoked Potentials based Brain-Computer Interface

by

Aravind Ravi

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2019

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see *Statement of Contributions* included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The publications that resulted from the work presented in this thesis:

A. **Aravind Ravi**, Sarah Pearce, Xin Zhang, and Ning Jiang "User-Specific Channel Selection Method to Improve SSVEP BCI Decoding Robustness Against Variable Inter-Stimulus Distance" (IEEE EMBS Conference in Neural Engineering 2019) [1].

B. **Aravind Ravi**, Jacob Manuel, Nargess Heydari, and Ning Jiang "A Convolutional Neural Network for Enhancing the Detection of SSVEP in the Presence of Competing Stimuli" (IEEE Engineering in Medicine and Biology Conference 2019) [2].

C. **Aravind Ravi**, Nargess Heydari, and Ning Jiang "User-Independent SSVEP BCI Using Complex FFT Features and CNN Classification" (IEEE Systems, Man and Cybernetics Conference 2019) [3].

D. **Aravind Ravi**, Nargess Heydari Beni, Jacob Manuel, and Ning Jiang "Comparing User-Dependent and User-Independent Training of CNN for SSVEP BCI" (Submitted Manuscript – Journal of Neural Engineering).

Contributions for the above publications: AR contributed to experimentation, data analysis, code development and writing of all the manuscripts. SP and XZ assisted in the experiment data collection for A. NH assisted in the data analysis for B, C, D and JM for B and D. NJ edited and provided directional guidance for this research.

Abstract

Non-invasive Brain-Computer Interfaces (BCIs) based on steady-state visual evoked potential (SSVEP) responses are the most widely used BCI. SSVEP are responses elicited in the visual cortex when a user gazes at an object flickering at a certain frequency. In this thesis, we investigate different BCI system design parameters for enhancing the detection of SSVEP such as change in inter-stimulus distance (ISD), EEG channels, detection algorithms and training methodologies.

Closely placed SSVEP stimuli compete for neural representations. This influences the performance and limits the flexibility of the stimulus interface. In this thesis, we study the influence of changing ISD on the decoding performance of an SSVEP BCI. We propose: (i) a user-specific channel selection method and (ii) using complex spectrum features as input to a convolutional neural network (C-CNN) to overcome this challenge. We also evaluate the proposed C-CNN method in a user-independent (UI) training scenario as this will lead to a minimal calibration system and provide the ability to run inference in a plug-and-play mode. The proposed methods were evaluated on a 7-class SSVEP dataset collected on 21 healthy participants (Dataset 1). The UI method was also assessed on a publicly available 12-class dataset collected on 10 healthy participants (Dataset 2). We compared the proposed methods with canonical correlation analysis (CCA) and CNN classification using magnitude spectrum features (M-CNN).

We demonstrated that the user-specific channel set (UC) is robust to change in ISD (viewing angles of 5.24°, 8.53°, and 12.23°) compared to the classic 3-channel set (3C - O1, O2, Oz) and 6-channel set (6C - PO3, PO4, POz, O1, O2, Oz). A significant improvement in accuracy of over 5% (*p*=0.001) and a reduction in variation of 56% (*p*=0.035) was achieved across ISDs using the UC set compared to the 3C set and 6C set.

Secondly, the proposed C-CNN method obtained a significantly higher classification accuracy across ISDs and window lengths compared to M-CNN and CCA. The average accuracy of the C-CNN increased by over 12.8% compared to CCA and an increase of over 6.5% compared to the M-CNN for the closest ISD across all window lengths was achieved.

Thirdly, the C-CNN method achieved the highest accuracy in both UD and UI training scenarios on both 7-class and 12-class SSVEP Datasets. The overall accuracies of the different methods for 1 s window length for Dataset 1 were: CCA: 69.1±10.8%, UI-M-CNN: 73.5±16.1%, UI-C-CNN: 81.6±12.3%, UD-M-CNN: 87.8±7.6% and UD-C-CNN: 92.5±5%. And for Dataset 2 were: CCA: 62.7±21.5%, UI-M-CNN: 70.5±22%, UI-C-CNN: 81.6±18%, UD-M-CNN: 82.8±16.7%, and UD-C-CNN: 92.3±11.1%.

In summary, using the complex spectrum features, the C-CNN likely learned to use both frequency and phase related information to classify SSVEP responses. Therefore, the CNN can be trained independent of the ISD resulting in a model that generalizes to other ISDs. This suggests that the proposed methods are robust to changes in inter-stimulus distance for SSVEP detection and provides increased flexibility for user interface design of SSVEP BCIs for commercial applications. Finally, the UI method provides a virtually calibration free approach to SSVEP BCI.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Ning Jiang, for providing me his constant support, guidance, valuable insights and patience throughout the journey of my Degree. I would also like to thank my thesis readers, Professor Shi Cao, and Professor James Danckert, for their valuable suggestions and feedback.

I would also like to thank my co-authors Xin, Nargess, Sarah, Yuanpei, Jacob and Prof. Jiang for their help in writing papers, data collection and conducting research.

I extend my gratitude to my well-wisher, Mr. Sastry Ramachandrula, for directing me in the right direction whenever needed and encouraging me to take up research.

I would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the University of Waterloo and the NSERC-CREATE program.

I extend my gratitude to the people who volunteered to take part in my study.

I would like to thank my parents, Mr. Ravi Soundararajan and Ms. Kalyani Ravi, for bringing me into this world and inspiring me throughout my life. I thank my sister, Aarthi Ravi, for her encouragement and support. I thank all my relatives and friends for their support.

I pay my obeisance to God, for bestowing upon me good health, strength, courage and inspiration.

Dedication

To my beloved parents and sister.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANOVA	Analysis of Variance
BCI	Brain Computer Interface
CCA	Canonical Correlation Analysis
CNN	Convolutional Neural Network
C-CNN	Complex Spectrum – Convolutional Neural Network
EEG	Electroencephalography
FFT	Fast Fourier Transform
ISD	Inter-Stimulus Distance
ITR	Information Transfer Rate
M-CNN	Magnitude Spectrum – Convolutional Neural Network
SNR	Signal-to-Noise Ratio
SSVEP	Steady State Visual Evoked Potential
t-SNE	t-Stochastic Neighborhood Embedding
UD/UI	User-Dependent/User-Independent

Chapter 1

Introduction

1.1 Brain-Computer Interfaces (BCI)

Brain-computer interfaces (BCIs) provide a direct pathway between the human brain and an external device for communication and control without the need of the conventional neuromuscular system [4]. BCIs can be especially useful for people with severe disabilities where the central nervous system is intact, but are unable to communicate due to damage in the peripheral nervous system or musculoskeletal system. In this case, a BCI can bypass the damaged pathways in order to provide the user with the ability to communicate or interact effectively with their surroundings. BCIs provide novel possibilities for neurorehabilitation for people with neurological disease such as stroke, amyotrophic lateral sclerosis (ALS) or paralysis [5]–[7]. A BCI system has the ability to capture, analyze and decode changes in the brain activity of the user, and translate them into useful commands to control/interact with an external entity. Figure 1.1 illustrates the five consecutive stages in a BCI system: signal acquisition, pre-processing, feature extraction, classification and control interface or application.

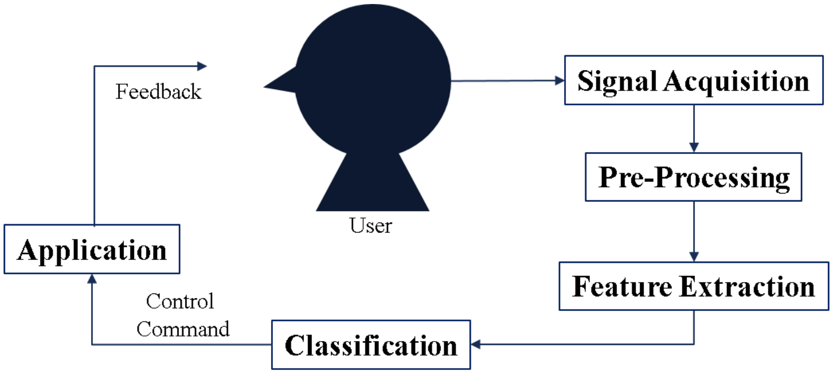

Figure 1.1 Stages in a Brain-Computer Interface system

1.1.1 Signal Acquisition

There are different methods used to record and monitor brain activity [8]. Brain imaging techniques such as functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS) can be used to monitor changes in metabolism known to be associated with changes in brain activity at specific parts of the brain. The neuronal activity can also be recorded and monitored by measuring the changes in the electrical activity of the brain with invasive techniques such as: intra-cortically with an array of electrodes or single electrodes, subdural from the cortex (also known as Electrocorticography or ECoG), or non-invasively i.e. directly from the scalp of the user (Electroencephalography or EEG). Another non-invasive method is based on measuring the magnetic activity by means of magnetic induction known as Magnetoencephalography (MEG). BCIs based on EEG are popular as they have the following desirable properties: non-invasiveness, high portability, high temporal resolution, ease of use, low-cost and few risks to users. The first attempt to develop an EEG based BCI system was by Vidal et. al. [9]. An EEG recording system consists of electrodes, amplifiers, analog-to-digital

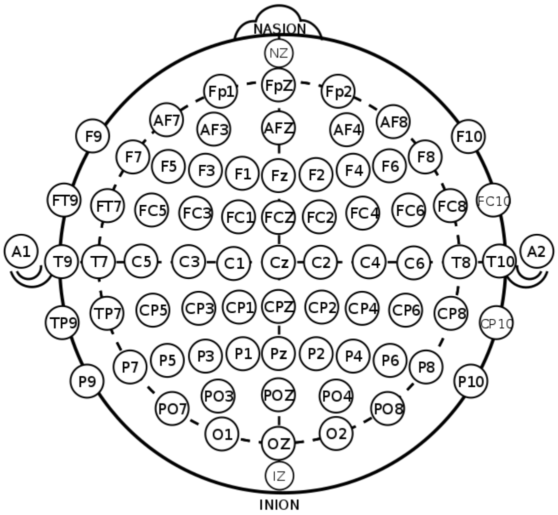

Figure 1.2 The International 10-20 system for EEG [68]

converters and a signal acquisition device. The EEG system measures the potential difference between a signal electrode and a reference electrode. An additional ground electrode is used to measure the differential voltage between the signal electrode and the reference electrode. These electrodes are usually coated with silver-chloride (AgCl) and a conductive gel is used to maintain good electrode-scalp contact (impedance usually between 1 kΩ and 10 kΩ). The electrodes are commonly placed at locations on the scalp based on the International 10-20 system (see Figure 1.2).

1.1.2 BCI Modalities

There are two categories of BCIs: (i) endogenous or exogenous BCIs [10] and (ii) synchronous or asynchronous BCIs. Endogenous BCIs allows the user to voluntarily modulate his/her neuronal activity based on intention. The BCI is based on spontaneously generated brain patterns. For example: Motor Imagery (MI BCI) [11] – the user imagines a motor movement such as moving the right hand/left hand or imagines lifting up the ankle (ankle dorsiflexion) [12]. These imagined movements elicit distinct responses in the EEG which can be detected and translated into a BCI command. Other types of endogenous BCIs include: Slow cortical potentials (SCP) based BCI, imagined tactile responses based BCI, etc. Exogenous BCIs are based on responses elicited due to an external stimulus. These responses are generated when the user focuses his/her attention on a stimulus which is associated with a BCI command. Examples include P300 BCI, steady-state visual evoked potentials (SSVEP) based BCIs, steady-state motion visual evoked potential (SSMVEP) BCIs [13], etc. Exogenous BCIs do not require extensive user training compared to endogenous BCIs. Furthermore, BCIs that require the user to have some control over their peripheral nerves and muscles (e.g. gaze control of the eyes) are called Dependent BCIs. BCIs that completely rely on volitional control and do not require the user to have peripheral control are called Independent BCIs. For example, SSVEP BCIs that depend on the user’s ability to fixate and focus on an external stimulus are called Dependent BCIs. Independent SSVEP BCIs based on user’s covert attention [14], [5], [6] have been proposed, but have lower performance compared to the former case. Synchronous and asynchronous BCIs are classified based on the input signal processing modality. Synchronous BCIs analyze the signals at pre-determined time windows and are usually tied to a cue. Therefore, the user is only allowed to send commands during these time periods and the signals outside the window are ignored. On the other hand, asynchronous BCIs continuously monitor and analyze the changes in the neuronal activity and offers a more natural mode of interaction [8]. P300 BCIs are tied to a cue/trigger corresponding to the stimulus and are therefore termed synchronous BCI. MI and SSVEP based BCIs do not necessarily need a cue/trigger and are examples of asynchronous BCIs [15].

1.1.3 Feature Extraction

The performance of a BCI is mainly influenced by the feature extraction and detection algorithm used to analyze and decode the neural responses. Most responses manifest as changes in the oscillatory activity of EEG and hence for BCI applications these features have been of primary interest. Many feature extraction algorithms have been proposed and studied in the literature. Most commonly used features are based on the changes in the power of the EEG signal at different frequency bands and time point features [16]. The power spectral density (PSD) is one such tool that allows capturing these changes. Changes in signal power arise due to different types of events. A decrease in the power of the signal due to an event is called event-related desynchronization (ERD) and an increase in power is called an event-related synchronization (ERS). For example, imagination or execution of a motor action leads to an ERD in the EEG between 8 Hz and 12 Hz (alpha band) over the primary motor cortex and an ERS in the beta band (12 Hz – 26 Hz). Other features for BCIs include band power values, event-related potentials (ERPs), magnitude spectrum features based on Fast Fourier Transform (FFT), time-frequency features (spectrogram) etc. Band-power features can be used to develop a simple two-state BCI switch. For example, an increase in the band power in the alpha band can be observed in the EEG recorded over the occipital region of the cortex due to eye closure. This power is higher compared to the eyes open condition and can be exploited to develop a BCI switch [17]. Feature extraction steps for BCIs usually involve a calibration/training stage to extract the features and are further used to train a machine learning model or classifier to detect the corresponding brain states.

1.1.4 Classification

Supervised classification algorithms such as Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM) are most often used in BCI applications [16]. A majority of MI-based BCIs have been shown to achieve high decoding performances with these techniques. Statistical learning methods such as Canonical Correlation Analysis (CCA) [18], [19] and its variants such as Filter bank CCA (FBCCA) [20], Individual Template-CCA (IT-CCA) [21], Combined-CCA [22], Combined-tCCA or Adaptive Combined CCA (A3C) [23] and Task-Related Components Analysis (TRCA) [24] have been used for SSVEP BCIs. Some of these methods have been evaluated in this thesis for comparison. Recently there has been increased interest in the application of neural networks and deep learning based algorithms to improve the overall classification performances of EEG based BCIs [25] [26], [27].

A simple feedforward neural network consists of an input layer, a hidden layer and an output layer with a number of nodes. In this network, the information from the data/features flows uni-directionally from the input to the hidden to the output layer. The goal of the neural network is to approximate some function that maps the input to a category/label. The network learns the value of the parameters that best approximate the function. The overall length of the chain of layers gives the depth of the model giving rise to the name *deep learning* [28].

Deep learning algorithms are a subset of machine learning algorithms in which the model is trained to learn both the features and the classifier directly from the data. Compared to classical machine learning algorithms that require a sophisticated feature extraction step as input to a classifier, deep learning methods offer the advantage of automatic feature extraction

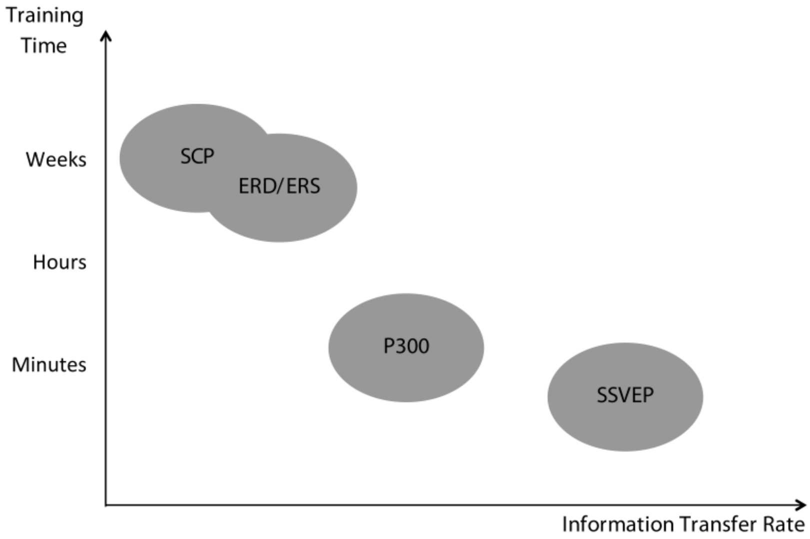

Figure 1.3 BCI Paradigms comparing training time and information transfer rate [34]

directly from the raw EEG data or in the transform domain of the EEG. A recent survey on deep learning for EEG indicated that 43% of studies used a specific type of deep learning algorithm known as the Convolutional Neural Network (CNN) for classification tasks [27]. CNNs are useful in cases where the data consists of a grid-like topology. For example: (i) EEG data - consists of a number of EEG channels and samples/time-series data, and (ii) Image data - consists of a 2-D grid of pixels. Recent studies showed that CNNs can significantly improve the classification performance in different types of BCI tasks such as motor imagery, P300 and movement-related cortical potentials (MRCP) [29] and SSVEP [30]–[33]. All these methods involve some form of calibration that requires the collection of training data to develop a classifier model and use the trained model in a real-time scenario. In addition to this, the user of a BCI system also requires training to regulate and elicit certain types of brain responses in order to use the system. This training is usually delivered through the means of neurofeedback [34]. The time required for a user to become trained varies depending on the BCI task. This is illustrated in Figure 1.3 [35] . Therefore, this challenge has to be considered while developing practical BCI applications.

Inter-subject variability is another challenge that has been reported in the BCI literature. This has been documented as “BCI illiteracy” in which it is estimated that 15% - 30% of the users cannot attain effective control with a BCI [36], [37]. To minimize the inter-subject variability and illiteracy problem, user-customized and user-dependent BCI designs have been proposed wherein the system parameters are calibrated for each individual user [37]. This has been shown as an effective solution to mitigate the effects of inter-subject variability [17]. More recently, BCI paradigms such as P300 and SSVEP have been considered as potential candidates to develop calibration-free or user-independent BCIs as they have consistent responses across most human subjects [38]. Compared to P300 BCIs which are synchronous in nature, SSVEP BCIs can operate asynchronously. SSVEP BCI has the properties of high signal-to-noise ratio, low participant training time, reduced number of EEG channels and high information transfer rate (ITR). In the next section, an overview of SSVEP BCI is provided and some of the challenges are explained.

1.2 Steady-State Visual Evoked Potentials (SSVEP)

When a user is presented with a visual stimulus, the light arrives at the photoreceptors in the retina and reach the retinal ganglion cells. The action potentials generated here propagate through the optic nerves to the visual cortex and other regions of the brain. These action potentials arise as a response to the visual stimulus [39]. Steady-state visual evoked potentials are responses elicited when a user focuses his/her attention on a repetitive visual stimulus (a light source) flickering at frequencies higher than 6 Hz [40]. These are periodic responses prominently observed in the occipital and occipito-parietal areas of the cortex. SSVEP responses appear as an increase in the amplitude of the signal at the fundamental frequency and its harmonics for the corresponding stimulus attended by the user. An example of the

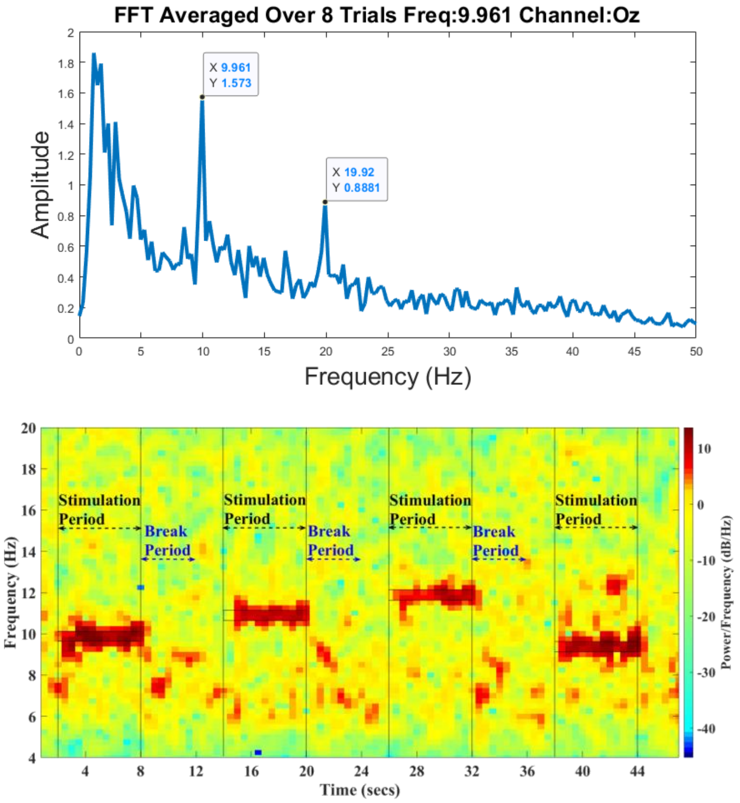

Figure 1.4. (a) Magnitude spectrum of the SSVEP averaged over 8 trials when the user was focusing on a stimulus with frequency 9.961 Hz. (b) Spectrogram of the first four trials of SSVEP on channel Oz.

magnitude spectrum of a typical SSVEP response averaged over eight trials is shown in Figure 1.4 (a) when the user was focusing on a stimulus with a frequency 9.961 Hz. Figure 1.4 (b) illustrates an example of the spectrogram of four consecutive trials of SSVEP signals at frequencies 9.961 Hz, 10.84 Hz, 11.87 Hz and 9.375 Hz collected over the channel Oz. Therefore, by analyzing the dominant frequency response in the EEG, the stimulus attended to by the user can be identified.

One of the commonly raised points of criticism for SSVEP BCIs is that they require shifting of the gaze from one visual stimulus to the other for control, and therefore, an eye-tracking system could track the gaze and achieve similar control. This has been studied in the literature under visual attention paradigms which have shown that humans have the ability to shift attention without shifting gaze. Some studies have shown that shifting gaze is not required for SSVEP BCIs, although doing this enhances the performance [14], [5], [6]. In [38], the authors discuss that although an eye-tracker could detect where a person is looking, it cannot determine if they are actively engaged in attending to the visual target. This could be achieved with sophisticated mechanisms and calibration of the tracker. Whereas, if there are multiple stimuli present in the same gaze direction, an SSVEP BCI could possibly be used to determine the visual target of interest [5].

The characteristics of the SSVEP response are influenced by the stimulus design parameters such as: frequency, color, shape and inter-stimulus distance (ISD). Based on the signal-to-noise-ratio (SNR), the flicker frequencies can be divided into three bands: (i) low-frequency – 5 Hz to 29 Hz (centered at 15 Hz), (ii) mid-frequency – 29 Hz to 37 Hz (centered at 31 Hz) and (iii) high-frequency – 37 Hz to 45 Hz (centered at 41 Hz) [41]. The low-band has been shown to produce higher SSVEP responses, in particular in the alpha band [41], [42]. Compared to other colors, a white color stimulus has been shown to elicit higher responses in SSVEP [43] [44]. Change in ISD has been shown to have a positive correlation with the performance of an SSVEP BCI. Closely placed SSVEP stimuli compete for neural representations leading to the effect of competing stimuli [39], [45], [1]. This challenge has been addressed in this thesis and has been detailed further in Section 1.3.

The visual stimulus presentation plays an important role in SSVEP BCIs [46]. These stimuli can be presented either through a set of blinking light emitting diodes (LEDs), on a computer screen (LCD displays) [47], or on virtual/augmented reality (VR/AR) displays [48]. LEDs have been shown to provide higher SSVEP responses compared to stimuli presented on a computer screen. But presenting the stimuli on an LCD/digital display offers the flexibility to easily configure the stimulus parameters such as frequency, shape, size, color, and inter-stimulus distance (ISD). Conventionally, the number of frequencies that can be presented on a computer monitor was limited by the screen refresh rate and therefore the frequencies that could be realized were factors of the refresh rate. In the conventional approach, the number of frames in each cycle remains constant. To produce a 10 Hz flicker, the frame sequence reverses between a black frame and a white frame every three frames on a screen with a refresh rate of 60 Hz. Based on this principle, the possible flicker frequencies that could be presented were: 6 Hz, 7.5 Hz, 8.57 Hz, 10 Hz, etc. In [42], the authors identified that the usable frequencies in the alpha band using this approach would only provide 8.57 Hz, 10 Hz and 12 Hz. Realizing a multiclass SSVEP with a frequency implementation based on factors of the refresh rate becomes challenging. As a solution to this problem, the authors proposed a novel frame based technique that allows realizing many flicker frequencies on an LCD monitor and overcomes the challenge of using the conventional method [42]. Furthermore, the authors showed that by approximating the presentation rate using a variable number of frames in each cycle, it was possible to realize 16 flicker frequencies between 9 Hz and 12.75 Hz with a frequency resolution of 0.25 Hz. Other studies have proved this as an effective method to generate visual flicker frequencies [39], [49]. This method has been used in this thesis to generate the visual flicker frequencies explained in Section 2.1.

Some of the other parameters that influence the performance of an SSVEP BCI system are the number of channels [50], feature extraction, detection algorithms and training methods. Increasing the number of EEG channels often leads to better classification performance in BCIs due to a higher number of features available for processing and classification. But this often leads to a cumbersome and long setup with many electrodes attached to the scalp of the user. In this thesis, we have selected six electrode locations in the occipital and parieto-occipital regions of the cortex to study the SSVEP responses. We also propose to use three channels among the six for ease of use and reduced setup time.

Several feature extraction and classification methods have been proposed for SSVEP processing. These methods can be classified into three broad categories: calibration free, user-dependent and user-independent methods [51]. Algorithms that are calibration free do not require any training data from the user and the system essentially becomes a plug-and-play type of interface for the user. One of the earliest methods in this category involves identifying the maximum of the sum of harmonics among the different stimulation frequencies calculated from the PSD of the SSVEP signal [52]. The most widely used calibration-free technique for SSVEP BCIs is CCA [18], [19]. CCA is a multivariable statistical technique that allows finding the underlying correlation between two sets of random variables. In the case of SSVEP, one variable is the SSVEP signal and the other is a set of sinusoidal reference templates. This method is commonly used as the baseline algorithm for SSVEP classification performance. This is explained in detail in Section 3.2.

User-dependent (UD) methods involve the collection of training data from each user and a model is developed based on user-specific features. As highlighted earlier, this method of user-based customization has been shown to overcome the inter-subject variability in BCIs. The UD methods developed for SSVEP classification have been extensions of the CCA algorithm. The most widely used UD methods are: Combination method-CCA [22], Individual Template and CCA (IT-CCA) [21]. And more recently proposed Task Related Components Analysis (TRCA) [24].

The user-independent (UI) methods involve the development of a model based on data collected from multiple users. This model is developed such that it can be applied to an unseen user, thereby virtually becoming calibration-free for the unseen user. There has been an increased interest to develop UI algorithms to improve the overall decoding performance of the SSVEP BCIs. Some of the UI methods developed are: Filter-Bank CCA (FBCCA) [20],

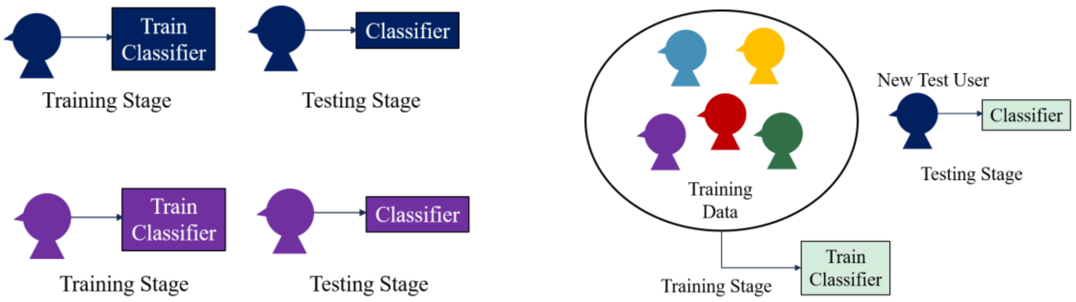

Figure 1.5. A diagram representing the User-Dependent (UD) and User-Independent (UI) training scenarios.

LASSO [53], Combined-tCCA and Adaptive Combined CCA (A3C) [23]. Figure 1.5 illustrates the UD and UI based training scenarios. Development of a UI CNN classification for SSVEP is one of the goals of this thesis and has been detailed in Section 1.3.2.

There has been a rising interest to apply deep learning algorithms for detection in SSVEP BCIs. CNNs have been the most frequently used deep learning technique for SSVEP BCIs. These methods have been proven to provide significant improvement in performance compared to traditional techniques [30]–[33], [54]. In, [30], [31], [33] the time domain SSVEP signals were transformed into the frequency domain before providing as input to the CNN for classification. In [30], the FFT was applied to the SSVEP signals and this representation of the signal was provided as input to the CNN. This was done as SSVEP responses usually manifest as an increase in amplitude at frequencies corresponding to the targeted stimulus. In [33], a similar frequency domain transformation was applied as input to a CNN to SSMVEP BCI to distinguish between Intentional Control (IC) and No Control (NC) state in an asynchronous mode. Next, CCA was used to classify the SSMVEP targets. The magnitude spectrum based approach along with the CNN was shown to outperform the traditional approaches such as CCA-threshold based method and CCA-KNN method. Similarly, FFT based transformation was applied to extract the magnitude spectrum of the SSVEP data in [30] and [31].

Furthermore, in [31], the authors showed that this method along with CNN classifier outperformed LASSO in decoding the SSVEP targets.

On the other hand, studies have attempted to provide the time domain SSVEP directly as input to a CNN for classification [32], [55]. When comparing the CNNs using time-domain inputs, a CNN using frequency-domain inputs would have a similar but relatively simple network structure, which means a relatively reduced number of trainable parameters and reduced computational complexity. One of the highly desirable properties for any deep learning based approach is to have a generic and simple architecture that works across multiple datasets. Challenges arise when using time domain as input to a CNN. The dimension of the time domain data is directly dependent on the sampling rate of the EEG acquisition system. Therefore, a CNN architecture developed based on this approach can vary significantly for each system. When there is a mismatch in the sampling rate, an up sampling or down sampling step maybe required before running inference on the developed model, and this could lead to loss of information. Another parameter that influences the model is the window length of the time domain data as this impacts the ITR of the system. When the window length changes, the input layer of the CNN is required to be modified. Therefore, we can address these challenges by fixing the resolution of the FFT and provide the frequency domain input to the CNN. Furthermore, all earlier studies using CNN for SSVEP classification were exclusively based on the magnitude spectrum of FFT and did not consider the phase related information [30], [31]. Using the phase information in SSVEP has been shown to improve the overall accuracy of the system [56], [57]. In this thesis, we propose a method to use the complex FFT representation directly as input to the CNN. This approach combines the real and imaginary parts of the FFT before providing as input to the CNN thus both the amplitude and phase related information in SSVEP can be extracted and used for decoding the targets. The proposed method has been explained in detail in Section 4.2.2.

In this thesis, we have addressed the challenge of competing stimuli and the development of a user-independent SSVEP BCI system. The following sections explain these challenges in detail and outline the solutions proposed in this thesis.

1.3 Motivation

1.3.1 Effect of Competing Stimuli

In SSVEP BCI, it is common to have multiple visual stimuli displayed simultaneously on a screen. When these stimuli are placed in close proximity within the visual field, all the stimuli elicit SSVEP. Selective visual attention has been shown as a property arising due to slow, competitive interactions that work in parallel across the visual field. Thus, objects in the visual field compete for limited visual processing capacity and control behavior [58]. Therefore, closely placed SSVEP stimuli compete for neural representations leading to the effect of competing stimuli [39], [45]. This effect has an influence on the resultant SSVEP response and on the performance of the BCI [39], [45], where a positive correlation was found between the overall accuracy and ISD. The findings in [45] were based on features extracted from the PSD on channels O1, O2 and Oz with SVM classifier. The authors suggested that further studies were needed to investigate better classification algorithms that would enhance the SSVEP decoding performance in the presence of significant competing stimuli. The authors also speculated that using electrode sites in the occipitoparietal region and classification methods like CCA could also influence the overall performance.

In this thesis, we studied the influence of changing ISD on the decoding performance of an SSVEP BCI. It is important to study this influence, because it limits the flexibility of the stimulus interface design for SSVEP BCIs by imposing a constraint on the acceptable inter-stimulus distance (ISD). As a consequence this limits the range of applicability of SSVEP for practical applications. Moreover, with increased interest in displaying SSVEP targets on a virtual/augmented reality interface [48] [59], this type of analysis would be beneficial as the visual stimuli would tend to be very closely spaced. Another limitation that has been addressed in this thesis is the required number of EEG channels. Two types of analyses were performed to investigate the effects of using certain EEG channels and classification algorithms due to change in ISD; outlined in Sections 1.4.1 and 1.4.2 respectively. The next Section details the need for a UI training method for SSVEP-BCIs.

1.3.2 User Independent BCI

UI BCIs provide great potential in many application scenarios where user-customized calibration is not feasible. These BCIs usually require no calibration or are pre-calibrated using minimal training data collected from multiple users and are able to run in a plug-and-play mode for the new unseen user. The SSVEP BCI modality elicits neuronal responses that are consistent across users and are most favourable for developing UI BCIs.

Due to the increasing number of studies using CNNs for SSVEP classification, it is necessary to evaluate and understand how CNNs perform in a UI training modality. Several studies using CNNs for SSVEP detection have reported the results for UD training. On the contrary, [32] was one of the earliest studies to evaluate a CNN based on UI training procedure for SSVEP BCI. The authors provided the pre-processed time domain SSVEP signal as input to the CNN and showed the ability of the CNN to classify twelve SSVEP targets among ten participants in a UI training scenario. Moreover, they showed that the network used the phase related information that aided in improved classification accuracy. Several of these studies have independently reported UD and UI based training results of CNN for SSVEP, but to our knowledge, only a few have compared the performances between the two approaches. In [51], the authors surveyed a number of training methods for SSVEP and highlighted that there was a glaring gap in the literature for a lack of comparative studies between UD and UI based training for SSVEP BCIs. Therefore, in this thesis, we have addressed this gap by providing a comparison of UD and UI training of CNN as part of Study III. Specifically, we compare the performance of different feature extraction methods with CNN for SSVEP classification.

1.4 Thesis Outline

The overall aim of this thesis is to study the influence of certain system design parameters on the decoding performance of an SSVEP BCI. Specifically we explore the effect of change in inter-stimulus distance (ISD) or competing stimuli and user-independent training of an SSVEP BCI. The remainder of the thesis is organized as follows: Chapter 2 describes the experimental protocol, stimulus design and data collection procedure. Chapters 3, 4 and 5 detail the analysis performed to address the challenges highlighted in this thesis. Chapter 6 provides a summary of the thesis and directions for future work. The analysis presented in Chapters 3, 4 and 5 are briefly described here:

1.4.1 **Study I**

In Chapter 3, we investigate the challenges arising due to change in inter-stimulus distance. A user-specific channel selection method is proposed to enhance the overall decoding performance of SSVEP BCI under influence of competing stimuli. Specifically, we assess the decoding performance using the CCA classification algorithm against variable ISD.

1.4.2 **Study II**

In Chapter 4, we investigate if the decoding performance can be improved in the presence of competing stimuli based on a novel detection algorithm. The Convolutional Neural Network (CNN) algorithm and two types of feature extraction methods were assessed for their robustness against variable ISD. We constrained the analysis to the classic 3-channel set used in the SSVEP literature i.e. O1, O2 and Oz. These results were compared with CCA as the baseline algorithm. This method used user-specific data for training and testing.

1.4.3 **Study III**

In Chapter 5, the proposed CNN architecture in Study II and the two types of feature extraction methods are evaluated and compared in UD and UI based training scenarios. We have also addressed a challenge related to reproducibility in deep learning based methods for BCIs reported in [26]. The authors provided guidelines to improve repeatability such as: clearly describing the architecture, providing a clear description of the data used, use of existing public datasets where possible and evaluating the performance with baseline. Therefore, the comparison of the proposed CNN and feature extraction methods were performed under both UD and UI training scenarios. CCA was used as the baseline algorithm. We have also performed this analysis on two datasets: (i) Dataset 1 - a seven class SSVEP dataset with 21 participants recorded in our lab and (ii) Dataset 2 - an existing twelve class SSVEP public dataset with 10 participants, which has been used by many earlier studies [22], [23], [32].

Chapter 2

Experiment

The seven class SSVEP dataset collected for this thesis was used in all the three studies listed in the outline (Section 1.4). Henceforth, this will be referred to as Dataset 1. The following sections describe the participant recruitment, experimental setup, SSVEP stimulus design, and data collection procedure for Dataset 1.

2.1 Participants

Twenty-one healthy adults (6 Females and 15 Males, aged 19-28 years) with normal or corrected-to-normal vision volunteered for the experiment. The experiment was approved by the Office of Research Ethics of the University of Waterloo (ORE # 31850). Written informed consent was signed by each participant before starting the experiment.

2.2 Experimental Setup

2.2.1 Stimulus Design

A total of seven stimuli were presented on an LCD display containing a refresh rate of 60 Hz. Each stimulus was associated with the following flicker frequencies: 8.42 Hz, 9.37 Hz, 9.96 Hz, 10.84 Hz, 11.87 Hz, 13.40 Hz and 14.87 Hz as illustrated in Figure 2.1. These frequencies were chosen based on prior studies in SSVEP [42], [60], as they have been shown to elicit higher amplitudes of SSVEP responses across most human subjects. For this experiment, the flicker frequencies were implemented based on the frequency generation

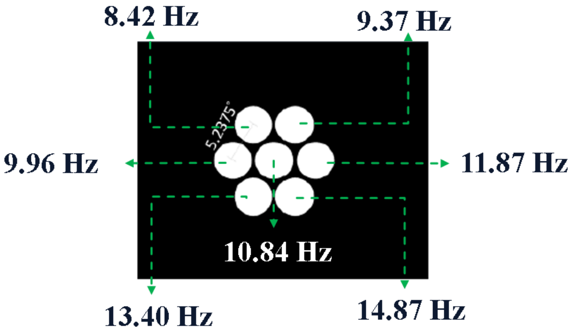

Figure 2.1 SSVEP stimulus design and flicker frequencies

techniques proposed in [61] and [49]. A sequence of ones and zeros were generated using this technique. A white frame and a black frame was displayed for a one and a zero in the sequence, respectively, resulting in a flicker pattern at the desired frequency. All stimuli were white in color and were chosen based on [43]. Circular targets have been shown to elicit higher responses compared to square [44]. Therefore, all stimuli were circular in shape.

To evaluate the effects of change in inter-stimulus distance (ISD), three different stimuli configurations were used (S_1, S_2 and S_3), which are illustrated in Figure 2.2. The ISDs are represented in units of viewing angles. The viewing angles are measured as a function of the distance between the centres of each stimulus and the eyes of the participant. One stimulus was placed at the centre of the screen and was aligned to the centre of the field-of-view of the

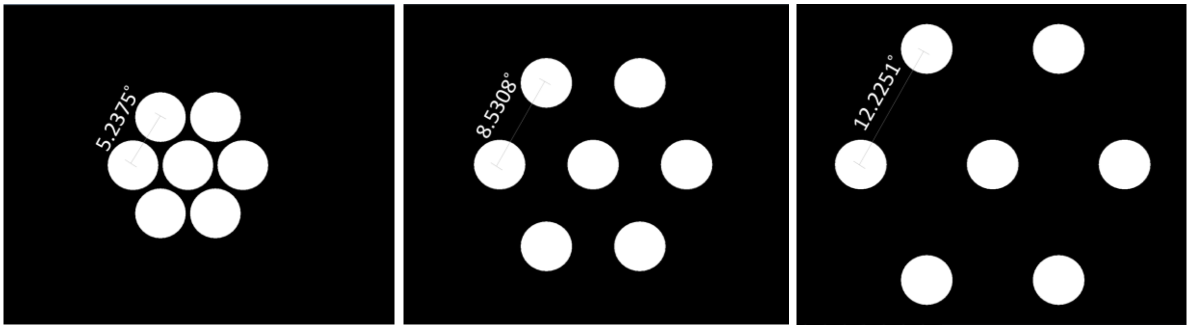

Figure 2.2. Stimulus Configurations - S_1, S_2, and S_3.

participant. Six surrounding stimuli were placed concentrically around the central stimulus. The viewing angles of the peripheral stimuli were 5.24°, 8.53°, and 12.23° (0.6 m between the eyes of the participant and the monitor). These are referred to as S_1, S_2 and S_3 respectively. The flickering stimuli have been designed and implemented in C++ on the OpenViBE platform [62].

2.2.2 Data Acquisition and Experimental Protocol

2.2.2.1 Data Acquisition

The EEG signals from each participant was acquired using the g.USBamp and Gammabox (g.tec Guger Technologies, Austria) wet electrode (g.Scarabeo) system with a sampling rate of 1200 Hz. Six active electrodes were used to record the EEG from the occipital and occipito-parietal areas according to the International 10-20 system. The electrodes O1, O2, Oz, PO3, POz and PO4 were used, FPz was used as the ground, and an electrode on the right ear lobe was used as the reference. The apparatus and electrode montage used for this experiment are shown in Figure 2.3.

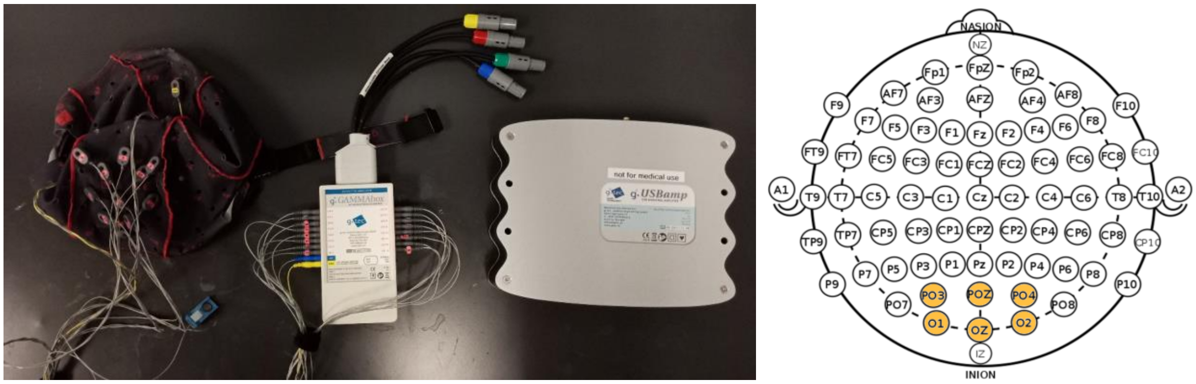

Figure 2.3. Data Acquisition hardware illustrating the EEG cap with the g.Scarabeo electrodes, the g.GammaBox driver and g.USBAmp Amplifier. 10-20 system based electrode montage depicting O1, O2, Oz, PO3, POz, PO4.

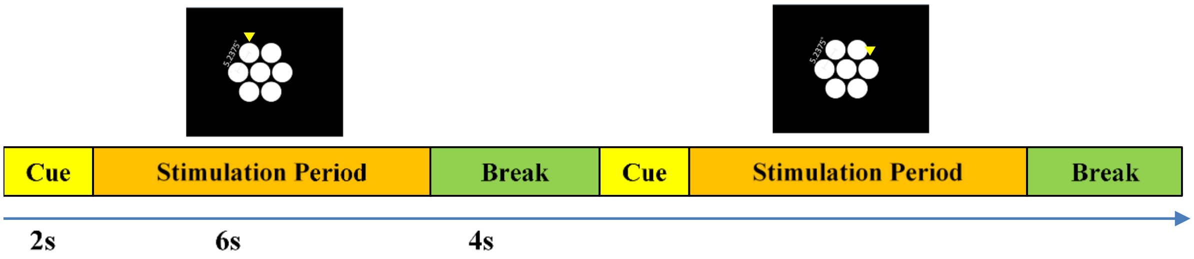

Figure 2.4. Experimental Protocol illustrating the 2 s cue period, 6 s stimulation period and 4 s break period for two consecutive trials.

2.2.2.2 Experimental Protocol

At the beginning of each trial, the participant was directed by a visual cue (inverted yellow triangle above the stimulus) to gaze at a specific stimulus. This cuing period was 2 s. A stimulation period of 6 s followed the cue period, during which the participant would focus on the targeted stimulus on screen for the entire duration. A 4 s break was provided before the start of the next trial. All seven flashing stimuli were presented simultaneously in every trial. One run consisted of 56 stimulus presentations. Each of the seven stimuli were presented eight times during the entire run. A total of three runs were performed, one for each stimulus configuration. Several minutes of resting period was provided between runs. The stimulus presentation sequence was randomized. For each participant, the order of the three runs was randomized. In addition, they were asked to avoid eye blinks or any sudden jerky movements during each trial. The experimental protocol is illustrated in Figure 2.4. The experimental protocol was designed in OpenViBE. All data were recorded, stored in the GDF format and was analyzed offline using MATLAB.

Chapter 3

Study I – User-Specific Channel Selection

In this study, a user-specific channel selection algorithm is proposed to overcome the effects of the competing stimuli in SSVEP BCI and is robust in performance [1]. This was studied under the three different stimuli configurations or ISDs.

3.1 Pre-Processing

All the signals were filtered using a 4th order Butterworth band-pass filter between 1 Hz and 40 Hz as this band consisted of the required SSVEP information. This step ensures that the noise due to DC drift and any high frequency noise were removed from the signal. The SSVEP data between 0.5 s and 3.5 s of each trial was used in the analysis. All six-channels were used for this analysis.

3.2 Canonical Correlation Analysis (CCA)

CCA is a multivariate statistical technique used to find the underlying correlation between two sets of multidimensional variables. It can be defined as the problem of finding two sets of basis vectors, one for x and the other for y, such that the correlations between the projections of the variables onto these basis vectors are mutually maximized [63]. Prior studies have shown that CCA can produce superior performance in detecting SSVEP responses in EEG [18], [19]. And most widely used as a baseline classification method for SSVEP detection [18], [20]–[22], [51], [64]. CCA is based on linear transformations. Consider the linear transformations $x = X^T w_x$ and $y = Y^T w_y$, where X refers to the multi-channel EEG data and Y refers to a set of reference signals of the same length as X. The objective of CCA was to find the projection vectors w_x and w_y that maximize the correlation between x and y by solving:

$$\rho(x,y) = max\frac{\mathrm{E}[xy]}{\sqrt{\mathrm{E}[x^2]\mathrm{E}[y^2]}} = max_{w_x,w_y}\frac{\mathrm{E}\left[w_x^T XY^T w_y\right]}{\sqrt{\mathrm{E}[w_x^T XX^T w_x]\mathrm{E}\left[w_y^T YY^T w_y\right]}}. \tag{3.1}$$

$$Y_n = \begin{bmatrix} sin(2\pi f_n t) \\ cos(2\pi f_n t) \\ . \\ . \\ sin(2\pi N_h f_n t) \\ cos(2\pi N_h f_n t) \end{bmatrix}, t = \left[\frac{1}{f_s}, \frac{2}{f_s}, \dots, \frac{N_s}{f_s}\right], \tag{3.2}$$

The reference signals Y_n were defined as (3.2), where $Y_n \in \mathbb{R}^{2N_h \times N_s}$, f_n was the stimulation frequency, f_s was the sampling frequency, N_s was the number of samples, and N_h was the number of harmonics. In this analysis, $N_h = 2$. The maximum of ρ with respect to w_x and w_y was the maximum correlation. The canonical features ρ_{fi}, where i =1, 2, …, K were extracted for each segment of the EEG data and the output class C for a given segment was determined as: *C = argmax (*ρ_{fi}*).* K denotes the total number of classes. The CCA code implementation is provided in Appendix A.3.

3.3 Channel Selection Algorithm

To investigate the influence of change in ISDs on the SSVEP classification performance different three-channel combinations were explored. Three channel combinations were chosen as this was ideal for ease of setup. The objective was to identify a three-channel set that provided an improvement in the overall classification accuracy and had the least amount of variation across different stimuli configurations; thereby being robust to the effect of competing stimuli. Therefore, a channel selection method was performed for each participant to find the three-channel set (among the six channels selected for this analysis) that satisfied the above conditions. The channel selection code implementation is provided in Appendix A.2.

There are 20 different three-channel combinations among the six channels (PO3, PO4, POz, O1, O2, Oz). First, the average accuracies A_c^p for each channel combination $c \in \{1,20\}$, across the three stimuli configurations, were calculated for each participant, defined as:

$$A_c^p = \frac{1}{3}\sum_{i=1}^{3}\left(a_i^{p,c}\right), \tag{3.3}$$

where $a_i^{p,c}$ denotes the accuracy for the ith stimuli configuration for the pth participant and cth channel combination. CCA was used to compute the accuracy $a_i^{p,c}$ for every c. Next, all the three-channel combinations A_1^p to A_{20}^p were ranked in the descending order based on the average accuracies A_c^p, and the top 20% were selected, resulting in four candidates of user-specific channel combinations, i.e., $|c| = 4$. Finally, from these candidates, the best three-channel set was chosen as the one with the least amount of variation v_c^p in performance across the three stimuli configurations. The performance variation v_c^p across different stimuli configurations was measured as the difference between the maximum and minimum accuracy for a given channel combination for each participant. This is defined as:

$$v_c^p = Max\left(a_1^{p,c}, a_2^{p,c}, a_3^{p,c}\right) - Min(a_1^{p,c}, a_2^{p,c}, a_3^{p,c}) \tag{3.4}$$

3.4 Statistical Analysis

The SSVEP decoding performances between the classical 3-channel set (3C), 6-channel set (6C) and user-specific channel set (UC) were compared. 3C consisted of O1, O2 and Oz, the 6C consisted of PO3, PO4, POz, O1, O2 and Oz, and UC was the best three-channel set derived from the proposed method. The mixed-effect model ANOVA was designed as follows: the response variable was the classification accuracy, the participant was a random factor, the fixed factors were channel set with three levels: 3C, 6C and UC; and stimuli configurations

(S_1, S_2 and S_3). The null hypothesis was that the mean classification accuracies were same for all channel sets. The secondary hypothesis was that the stimulus configuration or ISD had no effect on SSVEP decoding accuracy. A confidence interval of 95% was used.

The SSVEP decoding performance varies considerably across the different ISDs. Therefore, to investigate if this variation across the three ISDs was affected by the choice of the channel set, a mixed-effect model ANOVA was performed. The v_i^p, variation in accuracy across ISDs was the response variable. The participant was a random factor and the channel set was a fixed factor with three levels 3C, 6C and UC. The significance level was set as α=0.05.

3.5 Results

A user-specific channel selection method robust to change in ISD was proposed in this thesis. The UC set selected based on the proposed methodology for each participant is summarized in Table 3.1. The average classification accuracies and average variation across different ISDs using the UC sets were compared with the 3C Set (O1, O2, Oz) and 6C (PO3,

Table 3.1 The selected user-specific channel set

Participants	UC Channels	Participants	UC Channels
S01	PO3-O1-Oz	**S12**	PO4-O1-O2
S02	PO3-Oz-O2	**S13**	PO4-O1-Oz
S03	PO3-PO4-Oz	**S14**	PO4-O1-Oz
S04	PO3-Oz-O2	**S15**	PO4-O1-O2
S05	PO3-O1-Oz	**S16**	PO4-POz-O2
S06	PO4-Oz-O2	**S17**	POz-O1-Oz
S07	PO4-O1-Oz	**S18**	PO4-Oz-O2
S08	PO4-O1-Oz	**S19**	POz-O1-Oz
S09	PO4-Oz-O2	**S20**	PO3-PO4-Oz
S10	PO4-O1-Oz	**S21**	POz-O1-O2
S11	O1-Oz-O2		

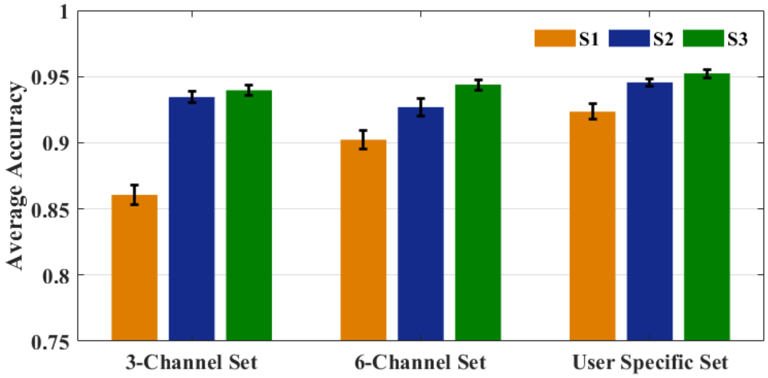

Figure 3.1. Average classification accuracies across all participants for each channel set and stimulus configuration.

PO4, POz, O1, O2, Oz). Figure 3.1 presents the average classification accuracies across all participants for each channel set and stimulus configuration S_1, S_2 and S_3. The results obtained for the 3C set are consistent with those in the literature that suggested that an increase in ISD lead to an increase in SSVEP decoding performance. The 6C set also showed a similar result in general, where the overall performance increases with increasing ISD. Compared to the 3C and 6C sets, the UC set improved the overall performance for all ISDs, particularly for S_1.

The mixed-effect model ANOVA revealed that the interaction between the ISD and the channel set had a significant effect on the classification accuracy (p=0.006). There was a significant effect of the channel set on the classification accuracy (p<0.001). Post-hoc comparison with Bonferroni correction found that the UC set produced a significantly better accuracy than the classic 3C set (p<0.001) and marginal significance than the 6C set (p=0.056). Specifically, there was a significant improvement in accuracy for S_1 with the UC: 92.34±7.5% compared to 3C:86.10±8.2% (p<0.001). There was no significant difference between: UC vs. 3C for S_2 and S_3 (p=1), and UC vs. 6C for S_2 and S_3 (p=1). There was no significant difference between the classic 3C set and 6C set (p=0.188).

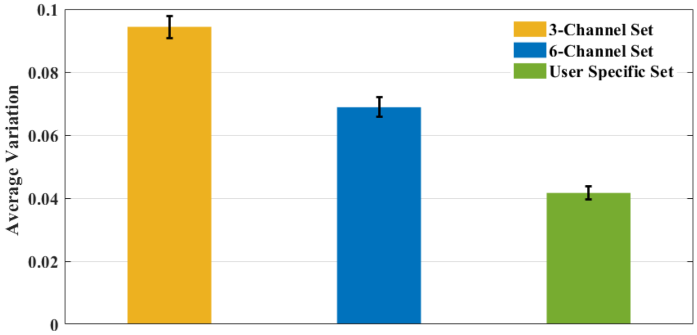

Figure 3.2. Average variation in performance across all participants for each channel set.

Figure 3.2 illustrates the average variation in performance between the 3C set, 6C set and the UC set. Evidently, the UC set showed the least variation in performance. The mixed-effect ANOVA revealed that the channel set had a significant effect on the performance variation across different ISDs (p<0.001). Post-hoc comparison with Bonferroni correction suggested that UC set produced significantly smaller variation in performance than 3C (p=0.035) and marginally smaller variation compared to 6C (p=0.053). Therefore, it can be concluded that the proposed user-specific channels selection can provide more robust SSVEP decoding accuracy when the ISD between the SSVEP stimuli varies.

3.6 Observations and Discussions

The results from the proposed channel selection method suggested that the UC set improved the CCA-based SSVEP decoding performance compared to the classic 3C set and 6C set setups, particularly for the spatially dense ISD (S_1). Figure 3.3 shows the relative frequency of the most commonly selected channels across participants. It can be observed that the channel Oz was selected for 80% of the participants along with PO4 and O1 selected for 60% of the participants respectively. The three-channel combination PO4, O1, Oz was selected in 5 participants out of the 21. Another observation was that in 95% of the participants the selected channels were a combination of both occipital and occipito-parietal electrodes. The

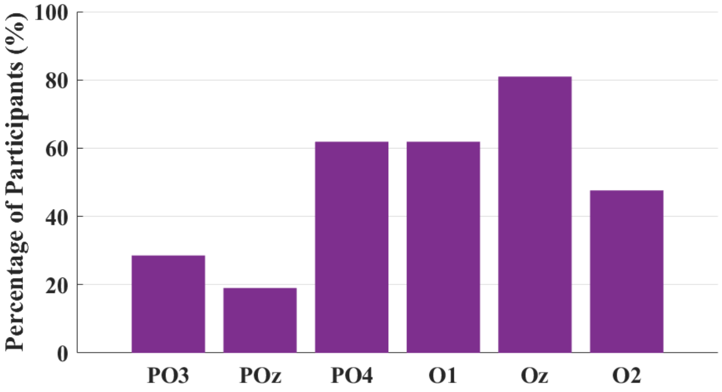

Figure 3.3. Most frequently selected channels across participants

average accuracy across ISDs increased by 5% using the UC set (94.5%) compared to the 3C set (89.5%). The largest improvement was observed in the S_1 which has significant effect of competing stimuli. It is interesting to note that there is no significant difference in performance between the UC and 6C set. This implies that with a reduced channel set of only 3 channels an improved performance can be achieved compared to using 6 channels. These results suggest that the UC set significantly reduced the effect of change in ISD. The average variation in performance reduced by 56% with the UC set compared to the classic 3C set (3C – 0.095 and UC – 0.041).

Therefore, the simple channel selection method allows a high level of SSVEP decoding performance and enhanced robustness against change in ISD. This provides enhanced flexibility in stimuli design and user-interface design for practical SSVEP applications. The proposed channel selection method requires only 3 s of data from each stimulus. This also provides a reduced amount of time for calibration (approximately 20 minutes of recording).

Chapter 4

Study II – CNN Robust to Change in ISD

In this study, we propose a Convolutional Neural Network (CNN) based classification scheme to enhance the overall SSVEP decoding performance in the presence of competing stimuli. Additionally, we investigate if the CNN is robust in performance to change in ISD with a reduced and fixed channel set.

4.1 Pre-Processing

All the signals were filtered using a 4th order Butterworth band-pass filter between 1 Hz and 40 Hz. The classical three channels set O1, O2 and Oz, most widely used in SSVEP BCIs, were used for this analysis. A data augmentation step was performed in addition to temporal filtering. Each 6 s trial was segmented based on sliding windows with different data lengths: 0.5 s, 1 s, 1.5 s, 2 s, 2.5 s, 3 s, and with a step size of 100 ms to increase the number of training epochs for the CNN.

4.2 Feature Extraction Methods

In this section, two types of feature extraction methods are detailed. Both methods transform the time domain EEG data into the frequency domain. The first method transforms the EEG data into the magnitude spectrum representation. The next method (proposed in this thesis) involves transforming the EEG signal into the complex spectrum representation. Both methods are particularly used as a feature extraction step applied to the EEG data before being fed as input to the CNN for classification.

4.2.1 **Magnitude Spectrum Features**

CNN has been used by prior studies for SSVEP classification [30], [31], [33]. These studies have used the magnitude spectrum features as input to the CNN. In the current analysis, this method is evaluated and compared with the proposed method (Section 4.2.2) for their ability to generalize and decode the SSVEP stimuli across different ISDs. In this method, the pre-processed time-domain EEG segments $x(n)$ were transformed into the frequency domain $X(k)$ computed by the Fast Fourier Transform (FFT). This resulted in a sequence of complex numbers $Re(X(k)) + jIm(X(k))$, from which the magnitude spectrum was calculated: $|X(k)| = \sqrt{Re(X(k))^2 + Im(X(k))^2}$. The frequency resolution of the FFT was fixed as 0.2930 Hz and the frequency components between 3 Hz and 35 Hz were selected. As a result, the length of the FFT transformed signal was $N_{fc} = 110$. The resultant feature vector computed along each channel were stacked together to form a matrix with dimensions $N_{ch} \times N_{fc}$, where N_{ch} was the number of channels and N_{fc} was the number of frequency components. Finally, this was provided as input to the CNN. In this analysis, we refer to this approach as the M-CNN method. An example of the input I_{M-CNN} for three EEG channels O1, O2 and Oz is defined as:

$$I_{M-CNN} = \begin{bmatrix} |FFT(x_{O1})| \\ |FFT(x_{Oz})| \\ |FFT(x_{O2})| \end{bmatrix} \tag{4.1}$$

This approach considers only the magnitude of the signals at different frequencies, but ignores the phase related information. Earlier studies in SSVEP have shown that the phase of the SSVEP signal contains significant information and provides improved decoding performance for SSVEP based BCI [32]. Therefore, in the next section, we propose a method that helps retain the phase related information from the complex spectrum representation of the SSVEP signals.

4.2.2 Complex Spectrum Features

The complex FFT representation was used to derive both magnitude and phase related information of a signal. First, the input time-domain signal was transformed into the complex FFT representation using the standard FFT with a resolution of 0.2930 Hz. Next, the frequency components of the real part and the imaginary part along each channel were extracted between 3 Hz and 35 Hz resulting in two vectors of length 110 each. These two vectors were concatenated into a single feature vector as: $\mathrm{I} = Re(X)||\mathrm{Im(X)}$, where the first half contained the real part and the second half contained the imaginary part of the complex FFT resulting in a vector of length 220. The resultant feature vector was stacked together to form a matrix with dimensions $N_{ch} \times N_{fc}$, where $N_{fc} = 220$. This approach of using the complex FFT as input to the CNN is referred to as the C-CNN method. An example of the input I_{C-CNN} is defined as:

$$I_{C-CNN} = \begin{bmatrix} Re\{FFT(x_{O1})\}, Im\{FFT(x_{O1})\} \\ Re\{FFT(x_{Oz})\}, Im\{FFT(x_{Oz})\} \\ Re\{FFT(x_{O2})\}, Im\{FFT(x_{O2})\} \end{bmatrix} \tag{4.2}$$

4.3 Convolutional Neural Network (CNN)

4.3.1 Network Architecture

The CNN architecture used in this analysis was inspired by the one proposed in [30]. Figure 4.1 illustrates the CNN architecture used in this analysis [2]. The CNN consists of four main layers, an input layer, two convolutional layers, and a fully connected output layer. The features extracted in the previous step were provided as input to the CNN. The input layer of the CNN, $I_{p.j}$, had dimensions $N_{ch} \times N_{fc}$, where $1 \leq p \leq N_{ch}$ and $1 \leq j \leq N_{fc}$. This was

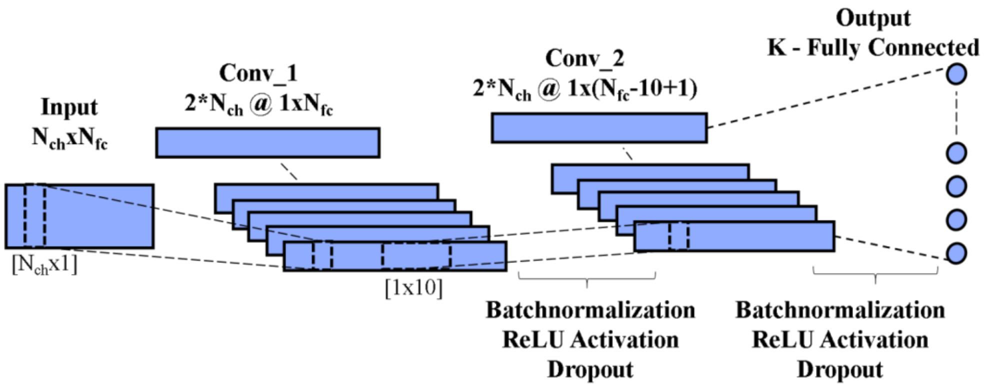

Figure 4.1. Convolutional Neural Network Architecture

followed by the convolutional layer Conv_1 was designed based on the spatial filtering concept. The kernel dimensions of $N_{ch} \times 1$ were used for this layer and performed 1D convolutions across the channel dimension (N_{ch}) of the input. The objective for this layer was to learn representations as a result of applying different weights to each channel in the input. The number of feature maps in the Conv_1 layer was $2 * N_{ch}$ with dimensions $1 \times N_{fc}$. The Conv_2 layer was designed to operate on the spectral representation dimension (N_{fc}) of the previous layer. The kernel dimension for this layer was 1×10. The number of feature maps in this layer were $2 * N_{ch}$. As a result of the convolution, the feature maps in this layer had the dimensions equal to $1 \times (N_{fc} - 10 + 1)$. Batch normalization was performed on the outputs of layers Conv_1 and Conv_2. Batch normalization has been shown to reduce the internal covariance within input samples resulting in the samples having zero mean and unit variance [65]. The rectified linear unit (ReLU) (4.4) was used as the activation function. To prevent overfitting, Dropout was added to the network as a regularization technique. Dropout and batch normalization have been shown to improve the generalization performance and training speed of neural networks [33], [65]. The number of units in the output layer (K) were kept equal to the number of SSVEP classes in the input data. The output layer was equipped with the *softmax* function to output the probability that a given input segment belonged to a particular class. The following section details the network learning procedure. The style of notation is similar to the one used in [30].

4.3.2 **Network Learning**

A unit in the network is defined by $x_k^l(p)$ where l is the layer, k is the feature map and p is the position of the unit in the feature map. Therefore, the output of a specific unit is defined as:

$$x_k^l(p) = f\big(BN(\sigma_k^l(p))\big), \tag{4.3}$$

where f is the ReLU activation function defined as:

$$f(\sigma) = max(\sigma, 0). \tag{4.4}$$

$\sigma_k^l(p)$ represents the scalar product of a set of input units and the weight connections between these units and the p^{th} unit in map k and layer l. BN represents the batch-normalization step. This will be explained in the subsequent sections. The computations performed by each layer are described as follows. The Conv_1 layer performs the following operation:

$$\sigma_k^1(p) = \sum_{j=1}^{N_{ch}} I_{p,j} w_k^1(j) + b_k^1 \tag{4.5}$$

where $w_k^1(j)$ is a set of weights with $1 \leq j \leq N_{ch}$ and b_k^1 is a bias. k indexes the feature map with $1 \leq k \leq 2 * N_{ch}$. In this layer, there are a total of $2 * N_{ch}$ feature maps and the size of the convolution kernel is $N_{ch} \times 1$. The output of each σ_k^1 is a $1 \times N_{fc}$ vector, where N_{fc} denotes the samples in the frequency domain. Next, batch normalization (BN) is performed on this output and passed through the ReLU activation function.

$$x_k^1 = f(BN(\sigma_k^1) \tag{4.6}$$

The output of Conv_1 layer, x_k^1 , is fed as input to Conv_2 layer.

$$\sigma_k^2(p) = \sum_{i=1}^{2*N_{ch}} \sum_{j=1}^{10} x_i^1(p+j-1)w_i^2(j) + b_k^2 \tag{4.7}$$

This layer is similar to Conv_1, the output of each σ_k^2 is a $1 \times (N_{fc} - 10 + 1)$ vector. BN and ReLU are applied to this output respectively. The output layer performs the following operation:

$$\sigma^3(p) = \sum_{k=1}^{2*N_{ch}} \sum_{i=1}^{N_{fc}-10+1} x_k^2 w_k^3(i) + b_k^3 \tag{4.8}$$

The softmax function is used as the activation function at the output layer:

$$S = \frac{\exp(x_p^3)}{\sum_{j=1}^{K} \exp(x_j^3)} \tag{4.9}$$

The output S is a vector of length K and each element in this vector corresponds to a probability score that a given sample belonged to a particular class.

4.3.3 **Training Parameters**

The weights of the CNN were initialized based on a Gaussian distribution $\sim N(0,0.01)$. The network was trained using the backpropagation technique by minimizing the categorical cross-entropy loss L:

$$L_i = \sum_{j=1}^{K} t_{i,j} log(S_{i,j}) \tag{4.10}$$

where $t_{i,j}$ is the ground truth probability score and $S_{i,j}$ is the predicted probability score for the i^{th} training example and j^{th} class. The stochastic gradient descent with momentum was used as the optimization algorithm to train the network. A grid search was employed as the hyper-parameter search strategy to find the best training parameters. The search space was defined as follows: Learning Rate $(\alpha) : \{10^{-3}, 2 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 10^{-1}\}$, Mini Batch size $(B) : 2^b;\ b \epsilon \{5, 6, 7, 8, 9, 10\}$, Dropout Ratio $(D) : \{0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, L2 Regularization $(L) : \{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}\}$, Number of Epochs $(E) : \{20, 30, 40, 50, 60\}$, and the ones that led to the best average accuracy across all participants were chosen. The hyper-parameter optimization was performed for both pipelines (M-CNN and C-CNN), separately. Within each pipeline, the same hyper-parameters were used for all participants and window sizes. The MATLAB Deep Learning Toolbox was used to implement the CNN and an example of the CNN code implementation is provided in Appendix A.1.

4.3.4 **ISD Independent Training Procedure**

The M-CNN and C-CNN methods were used in two scenarios. The first scenario involved training the CNN with the objective that it can generalize across different ISDs and can perform inference independent of the ISD. The second scenario involved comparing user-independent and user-dependent training of the CNN model (Section 5.2). The respective training procedures are explained in the following sections.

The M-CNN and C-CNN methods were evaluated in an offline manner. The objective of the analysis was to investigate whether model parameters learned on one ISD can generalize across other ISDs and sessions. Therefore, three cases were assessed. Case 1: CNN trained on S_1 and tested on S_2 and S_3; Case 2: CNN trained on S_2 and tested on S_1 and S_3; and Case 3: CNN trained on S_3 and tested on S_1 and S_2. For all cases, data ($S_{1,}$ S_2 and S_3) from the same participant was used for training and validation. As a result, for each stimulus configuration, two test-set accuracies were computed, and the mean accuracy was calculated for each participant. For example, test-set accuracy for S_1 was calculated from Case 2 and Case 3, and the mean of the two test accuracies was calculated. This training procedure was performed separately for each window size *W*.

4.4 Statistical Analysis

In this analysis, the performance of M-CNN and C-CNN were evaluated for their ability to classify the SSVEP and generalize across different ISDs. Both methods were compared with CCA as the baseline. The accuracies were calculated for each stimulus configuration and the overall performance was analysed using a mixed-effect model ANOVA. The classification accuracy was the response variable. The participant was a random factor, the window length (*W*) was a random factor with six levels (*W*=[0.5 s, 3 s]), the ISD was a fixed factor with three levels (S_1, S_2 and S_3); the classification algorithm was a fixed factor with three levels (CCA, M-CNN and C-CNN). The null hypothesis was that the average classification accuracies were the same for all algorithms. The secondary hypothesis was that the ISD has no effect on the performance. The significance level was set as α=0.05 for all analyses with adjusted p-values reported for post-hoc analyses.

4.5 Results

The objective of the analysis of the CNN based methods in this experiment was to investigate whether the CNN model parameters learned on one ISD can generalize across other ISDs and sessions. Figure 4.2 presents the average classification accuracy across all participants for the different classification methods (CCA, M-CNN and C-CNN), and ISDs S_1, S_2 and S_3, for data lengths of $W = \{0.5\text{ s}, 1\text{ s}, 1.5\text{ s}, 2\text{ s}, 2.5\text{ s}, 3\text{ s}\}$. The C-CNN outperformed M-CNN and CCA for all ISDs. The mixed-effect model ANOVA revealed a significant difference between all three algorithms and ISDs (p<0.001). There was a significant interaction between the classification algorithm and ISD (p=0.002). Post-hoc comparisons with Bonferroni simultaneous comparisons indicated that the C-CNN and M-CNN obtained a significantly higher accuracy on average than CCA across all ISDs (C-CNN: 92.3% vs. CCA: 80.9%; p<0.001 and M-CNN: 85.4% vs. CCA: 80.9%; p=0.009). Further analysis was performed to compare the improvement in performance between classification algorithms

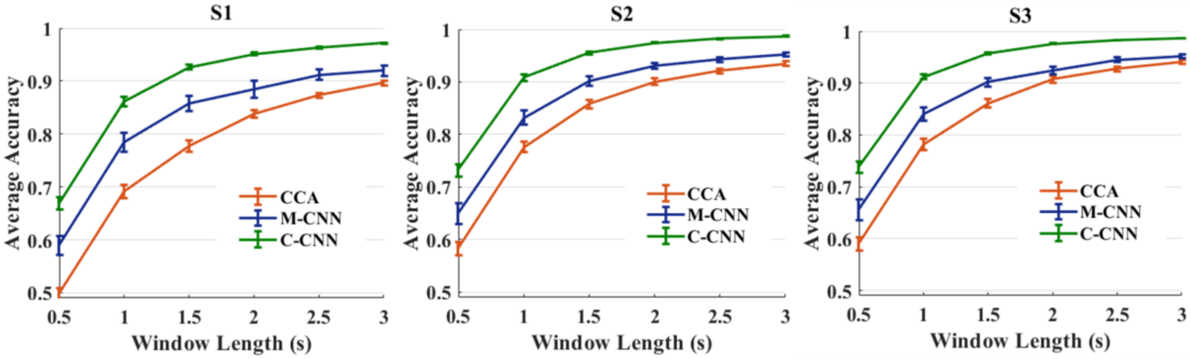

Figure 4.2. Comparison of the average accuracies across all participants for the different classification methods and data lengths of $W = \{0.5\text{ s}, 1\text{ s}, 1.5\text{ s}, 2\text{ s}, 2.5\text{ s}, 3\text{ s}\}$ for inter-stimulus distances: S_1, S_2, S_3. The vertical line overlaying each bar indicates the variance across participants.

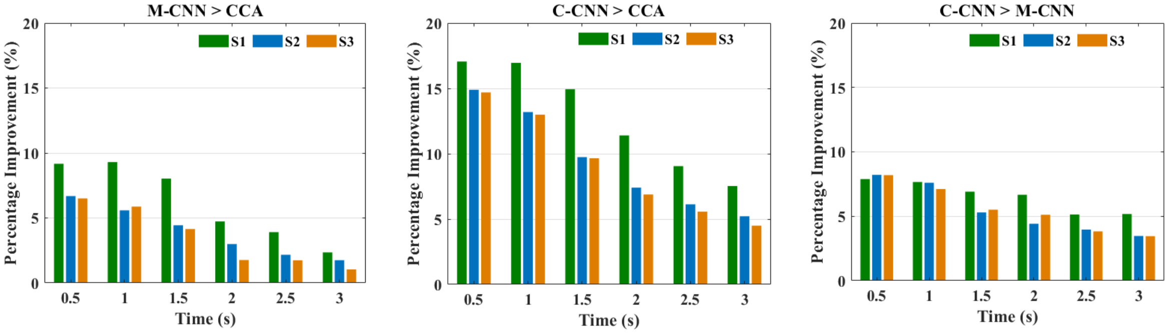

Figure 4.3 Comparing the percentage improvement across algorithms and ISDs. '>' operator indicates that the classification accuracy of the algorithm on the left of the operator was greater than the algorithm on the right of the operator.

across different window lengths and ISDs. Figure 4.3 presents this comparison. There was a significant improvement in performance between M-CNN vs. CCA for $W = [0.5\text{ s} - 1.5\text{ s}]$ ($0.001 \leq p \leq 0.010$). For $W \geq 2$ s, there was no significant difference in performance ($p \geq 0.59$). The comparison between C-CNN and CCA revealed significant improvement in performance across all window lengths ($p \leq 0.002$). It is also evident from Figure 4.3, an increase of over 12.81% can be observed on average for the closest ISD (S1) and for $W \leq 2$ s. Between C-CNN vs. M-CNN, the C-CNN was significantly higher for $W = [0.5\text{ s} – 2\text{ s}]$ ($p \leq 0.009$) and was not significant for $W \geq 2.5$ s ($0.054 \leq p \leq 0.088$).

4.6 Observations and Discussions

This analysis measured the performances of two types of feature extraction methods and CNN-based classification with the following objective: to enhance the decoding performance of an SSVEP-based BCI in the presence of competing stimuli with variable ISDs. The performance was compared with the CCA as the baseline method. The presented results indicated that the C-CNN is robust in decoding SSVEP across different ISDs and achieved the highest performance compared to M-CNN and CCA based methods. The average accuracy of the C-CNN increased by over 12.8% compared to CCA and an increase of over 6.5% compared to the M-CNN for the closest ISD (S1) across all window lengths. S_1 is the most challenging case with the most significant completing stimuli. The results also revealed that the proposed C-CNN method achieved significant improvements at shorter window lengths. This is particularly suitable for BCIs that require a higher ITR leading to faster detection rate.

From the results, it can be observed that the proposed CNN model with the complex spectrum features as input can be trained independent of the ISD. This results in a model that can generalize to new ISDs and sessions that were not seen in the training data. Furthermore, these results are suitable for practical applications as it is feasible to implement a calibration method with the ISD as discussed in Case 1. For example, a model can be trained with the calibration data obtained from visual interfaces with the closely spaced stimulus (smaller ISDs), and the model can run inference on interfaces with larger ISDs. These are particularly favorable for SSVEP based BCI applications developed on virtual reality or augmented reality interfaces where the on-screen stimuli would be very closely spaced. Therefore, from an interface design perspective, the proposed model provides more flexibility. For example, the application can be easily modified with a simple software update to accommodate newly configured stimulus distances and retain the same CNN model weights for inference.

Chapter 5

Study III – Comparing UD and UI Training of CNN

In this study, the UD and UI based training of CNN were compared. The following analysis was performed on the S_1 ISD of the 7-class SSVEP dataset (Dataset 1) as this was the configuration with significant competing stimuli. A 12-class SSVEP public Dataset (Dataset 2) was also used to evaluate the performance of the proposed CNN methods for their ability to generalize to other datasets. The analysis presented in this section are portions presented in manuscript D (*Statement of Contributions*).

5.1 12-Class SSVEP Public Dataset

An offline SSVEP dataset was downloaded from a public repository [22]. This dataset was collected on ten healthy volunteers and was used in many earlier studies [22], [23], [32]. All participants were seated in a comfortable chair at 0.6 meters from an LCD monitor in a dim room. Twelve flickering stimuli were displayed on the screen with the following flicker frequencies: 9.25 Hz, 9.75 Hz, 10.25 Hz, 10.75 Hz, 11.25 Hz, 11.75 Hz, 12.25 Hz, 12.75 Hz, 13.25 Hz, 13.75 Hz, 14.25 Hz, and 14.75 Hz. The stimuli were 6 cm x 6 cm squares that were arranged in a 4x3 grid and represented a numeric keypad.

The BioSemi ActiveTwo EEG (Biosemi B.V., Netherlands) system with a sampling rate of 2048 Hz was used to acquire the EEG data. Eight active electrodes were placed over the occipito-parietal areas. At the beginning of each trial, the participant was directed by a red square cue to gaze at a specific stimulus. This cuing period was 1 s. A stimulation period of 4 s was followed by the cue period and the participant was asked to focus on the targeted stimulus for the entire duration. One block consisted of 12 trials with one trial for each of the 12 stimuli on the screen presented in random order. A total of 15 blocks were presented leading to a total of 180 trials.

5.1.1 Pre-processing

Dataset 1 was pre-processed as discussed in Section 4.1. The signals from Dataset 2 were pre-processed based on [22] and [32]. All eight channels were used from this dataset. Temporal filtering was performed using a 4th order Butterworth band-pass filter between 6 Hz and 80 Hz. Each 4-second trial was divided into 1 s non-overlapping segments as per [32].

5.2 Training Procedure

The M-CNN and C-CNN methods discussed in Chapter 4 were evaluated in two training scenarios in this Section. A user-dependent training and a user-independent training scenario were evaluated. The respective training procedures are explained in the following sections. The training parameters were selected based on the same strategy as discussed in Section 4.3.3.

5.2.1 User-Dependent Training Procedure

A user-dependent model (M-CNN or C-CNN) was developed as follows: data from a single participant was used to train a model and was validated using the same participant's data. To achieve this, a 10-fold cross-validation procedure was performed. Pre-processing was carried out on all trials of a single participant using different window lengths (*W*) and both magnitude and complex spectrum features were extracted. Ten non-overlapping parts were generated from the pre-processed epochs and for each window length, the CNN was trained separately on nine parts and validated on the one remaining part. This method was carried out for Dataset 1. Similarly, Dataset 2 was split based on 10-fold cross-validation procedure on the 1 s segments of the epochs. For the purpose of direct comparison with earlier studies [22], [32], no other window length was used because 1 s was the window length used in those studies. The methods using UD training and magnitude spectrum features were referred to as UD-M-CNN and using UD training with complex spectrum features were called UD-C-CNN. The total number of 1 s segments in the training fold were: 2470 (Dataset 1) and 648 (Dataset 2) and testing fold were 274 (Dataset 1) and 72 (Dataset 2) respectively. The final parameters of the network were chosen as: $\alpha = 10^{-3}$, $momentum = 0.9$, $D = 0.25$, $L = 10^{-4}$, $E = 40$, $B = 256$ (Dataset 1), and $E = 50$, $B = 64$ (Dataset 2).

5.2.2 **User-Independent Training Procedure**

A User-Independent (UI) training procedure based on the magnitude and complex spectrum features (M-CNN and C-CNN) is proposed in this thesis. As mentioned previously, the UI training provides the ability for a model to be used in a plug-and-play mode. This means the model is trained on a pooled dataset containing multiple participants and can classify the data of an unseen user, leading to a calibration-free system. To achieve this, a leave-one-participant-out validation was used. If a given dataset contains N participants, then the model was trained on the data of N-1 participants and tested on the data of the unseen participant. This procedure was performed individually for both types of features (M-CNN and C-CNN) and for every W. For example, the total number of 1 s segments in the training folds were: 54880 (Dataset 1) and 6480 (Dataset 2), and testing folds were 2744 (Dataset 1) and 720 (Dataset 2) respectively. The parameters that resulted in the highest average accuracy across all participants were selected. These methods were referred to as UI-M-CNN and UI-C-CNN respectively. The final parameters of the network for Dataset 1 were chosen as: $\alpha = 10^{-3}$, $momentum = 0.9$, $L = 10^{-4}$, $D = 0.25$, $E = 50$, $B = 1024$ (C-CNN), and $B = 512$ (M-CNN). For Dataset 2, $\alpha = 10^{-3}$, $momentum = 0.9$, $D = 0.25$, $E = 50$, $B = 256$, $L = 10^{-3}$ (M-CNN) and $L = 5 \times 10^{-3}$ (C-CNN).

5.3 Statistical Analysis

In this analysis, statistical tests were performed to compare the performances of the UD and UI based training methods with the baseline CCA method. A mixed-effect model ANOVA was used to evaluate the classification methods. These were evaluated on both datasets, Dataset 1 and Dataset 2. The overall accuracy of each method was the metric of interest. Therefore, the response variable was the classification accuracy. The window length (*W*) was a random factor with six levels (*W*=[0.5 s, 3 s]), the participant was a random factor, and the classification algorithm was a fixed factor with five levels (CCA, UD-M-CNN, UD-C-CNN, UI-M-CNN, UI-C-CNN) respectively (Dataset 1). The null hypothesis was that the classification accuracy was the same for all classification algorithms. A 95% confidence interval was used for comparison and analysis. The same statistical analysis was performed on both datasets with slight modifications for Dataset 2, the window length was fixed as $W = 1$ s for the purpose of comparison with previous studies using Dataset 2. Therefore, it was not considered as a factor.

5.4 Results

5.4.1 **Dataset 1**

Figure 5.1 illustrates the classification accuracies across 21 participants for Dataset 1 of all the methods at different window lengths. The algorithms can be ordered from highest to lowest classification accuracy as follows: UD-C-CNN, UD-M-CNN, UI-C-CNN, UI-M-CNN and CCA. The UI-C-CNN achieved higher accuracies than UI-M-CNN and CCA among the UI methods. Similarly, the UD-C-CNN achieved higher accuracies than UD-M-CNN and CCA among the UD methods. The mixed-effect model ANOVA revealed a significant difference between all the classification algorithms (p<0.001). Post-hoc comparisons with Bonferroni simultaneous tests indicated that there was a significant improvement in performance using

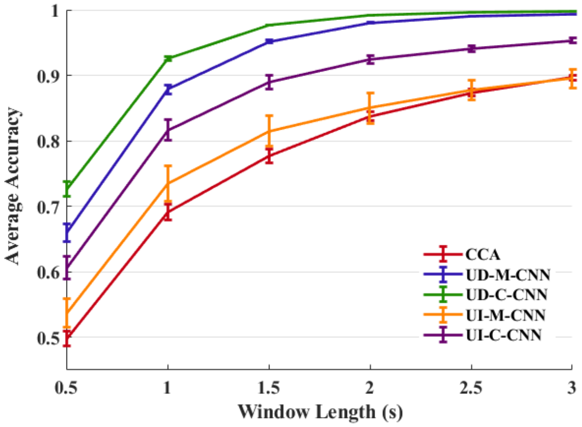

Figure 5.1. Dataset 1 - Comparison of the accuracies across all participants for the different classification methods for data lengths of *W* = {0.5s, 1s, 1.5s, 2s, 2.5s, 3s}. The vertical bars indicate the variance among the participants at each *W*.

UD-M-CNN, UD-C-CNN, and UI-C-CNN when compared to CCA (p<0.001). There was a significant difference between UI-M-CNN and UI-C-CNN (p<0.001). There was no significant difference between UI-M-CNN and CCA (p=0.729). There was no significant difference between UD-M-CNN and UD-C-CNN (p=0.386). Further analysis was performed to compare between the UD and UI methods based on each feature extraction technique; all comparisons were statistically significant: UI-C-CNN vs. UD-C-CNN (p<0.001), UI-M-CNN vs. UD-M-CNN (p<0.001), UD-C-CNN vs. UI-M-CNN (p<0.001), UD-M-CNN vs. UI-C-CNN (p=0.002). These results indicate that the proposed C-CNN method outperformed the M-CNN method in both UI and UD training scenarios. One advantage is that even when the C-CNN was used in the UI training scenario, it performed similarly to M-CNN used in the UD training scenario.

The interactions between window lengths and the different classification methods revealed that across all window lengths, both the UD methods outperformed CCA (p<=0.001). Comparing the UI methods, UI-C-CNN provided significant improvement than CCA for

W=[0.5 s, 2 s] ($0.004 <= p < 0.016$). Across all windows, UD-M-CNN was significantly better than UI-M-CNN ($p <= 0.002$). There was a significant difference in accuracy at lower windows from 0.5 s – 1.5 s between UD-C-CNN and UI-C-CNN ($0.003 <= p < 0.029$). The overall accuracies of the different methods for 1 s window length were: CCA: 69.1±10.8%, UI-M-CNN: 73.5±16.1%, UI-C-CNN: 81.6±12.3%, UD-M-CNN: 87.8±7.6% and UD-C-CNN: 92.5±5%. It can be inferred that both the UD training based methods have outperformed the UI training based methods and CCA. In particular, the C-CNN method achieves the highest accuracies in both training scenarios.

5.4.2 Dataset 2

Figure 5.2 summarizes the accuracies of all the classification methods for Dataset 2 across 10 participants for the *W*=1 s. It can be inferred that the UD-C-CNN method achieves the highest accuracy of 92.3±11.1%. Among both training scenarios, the UD methods outperform the UI methods and CCA, as expected. The average accuracies of the different

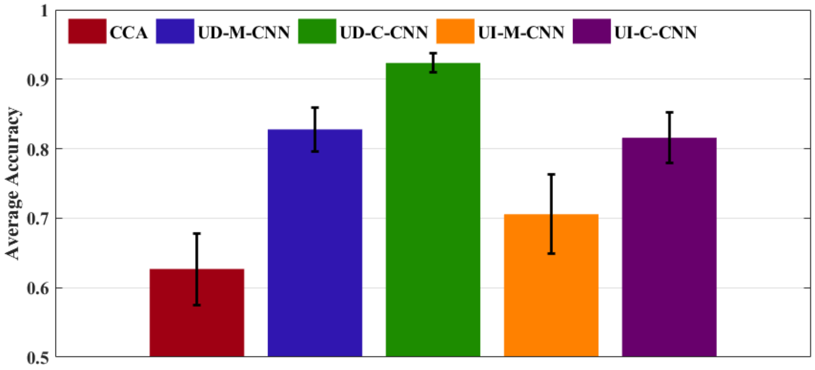

Figure 5.2. Dataset 2 - Comparison of the average accuracies across all participants for the different classification methods for data lengths of *W* = 1 s. The vertical line overlaying each bar indicates the variance among all participants in each method.

methods for data length of 1 s were: CCA: 62.7±21.5%, UI-M-CNN: 70.5±22%, UI-C-CNN: 81.6±18%, UD-M-CNN: 82.8±16.7%, and UD-C-CNN: 92.3±11.1%.

The mixed-effect model ANOVA revealed a significant difference between all the classification methods (*p*<0.001). Post-hoc Bonferroni simultaneous comparison was performed to compare the different algorithms. The UD-C-CNN, UD-M-CNN and UI-C-CNN significantly outperformed CCA (*p*<0.001). There was no significant difference between UI-M-CNN and CCA. There was a significant difference between UI-M-CNN and UI-C-CNN (*p*=0.016). There was marginal significance between UD-M-CNN and UD-C-CNN (*p*=0.054). Further analysis was carried out to compare between the UD and UI methods based on each feature extraction techniques: UI-M-CNN vs. UD-M-CNN (*p*=0.006), UI-C-CNN vs. UD-C-CNN (*p*=0.020), UD-C-CNN vs. UI-M-CNN (*p*<0.001). The difference between UD-M-CNN vs. UI-C-CNN (*p*=1) was not significant.

5.4.3 **Computational Load Analysis**

A computational load analysis was performed on both types of CNN models. The MATLAB Deep Learning Toolbox was used to implement the CNN and was trained on an Intel Core i5-8400 CPU @ 2.80 GHz and 8 GB RAM. The total number of trainable parameters for the Dataset 1 were: UD-M-CNN and UI-M-CNN = 4663. And for UD-C-CNN and UI-C-CNN = 9283. The overall training time to train 1 s segments were: UD-M-CNN: 6 s, UD-C-CNN: 12 seconds, UI-M-CNN: 3 min. 20 s and UI-C-CNN: 7 min. 17 s. For Dataset 2, the total number of trainable parameters were: UD-M-CNN and UI-M-CNN = 22188. And for UD-C-CNN and UI-C-CNN = 43308. The overall training time to train 1 s segments were: UD-M-CNN: 6 s, UD-C-CNN: 10 s, UI-M-CNN: 53 s and UI-C-CNN: 1 min. 50 s; the number of training samples was 6480.

5.5 Observations and Discussions

It can be observed from the results that the UD method outperforms the UI methods. An interesting observation was that the C-CNN methods performed better than the other methods in both UD and UI training scenarios. The UI-C-CNN achieved similar performance compared to the UD-M-CNN method and achieved higher accuracy than the UI-M-CNN method. To understand this further, we investigated the results of UI-M-CNN and UI-C-CNN by visualizing the learned features representations of the CNN on both datasets. The t-Stochastic Neighborhood embedding (t-SNE) technique was used to visualize the features [66]. This method has been widely used for the purpose of feature visualization. It enables the visualization of high-dimensional features into 2 or 3 dimensions [23], [32], [67]. Therefore, we visualized the features extracted from the input layer, the Conv_1_ReLU and the Conv_2_ReLU layers. For both datasets, the magnitude and complex spectrum features of 1 s long SSVEP segments were visualized.

5.5.1 Dataset 1

Figure 5.3 (a-f) illustrates the features extracted at different layers of the network for a participant's SSVEP data unseen by the classifier. (a-c) illustrate the features of the UI-M-CNN and (d-f) illustrate the features of the UI-C-CNN methods respectively. This was achieved by training the CNN on the datasets of N-1 participants and forward propagating the data of the unseen participant through the pre-trained network. As a result, the features were extracted at the output of each layer. In the illustration, each data point represents a 1 s segment of a single trial of SSVEP. The clusters are colored based on the class label. As we progress into the deeper layers of then network, we observe more and more clustered representations. The clustering in (c) and (f) represent the outputs of the Conv_2_Relu of M-CNN and Conv_2_ReLU of C-CNN. From the results of the C-CNN, a better class separation and clustering can be observed compared to M-CNN. This result is likely the outcome of using the

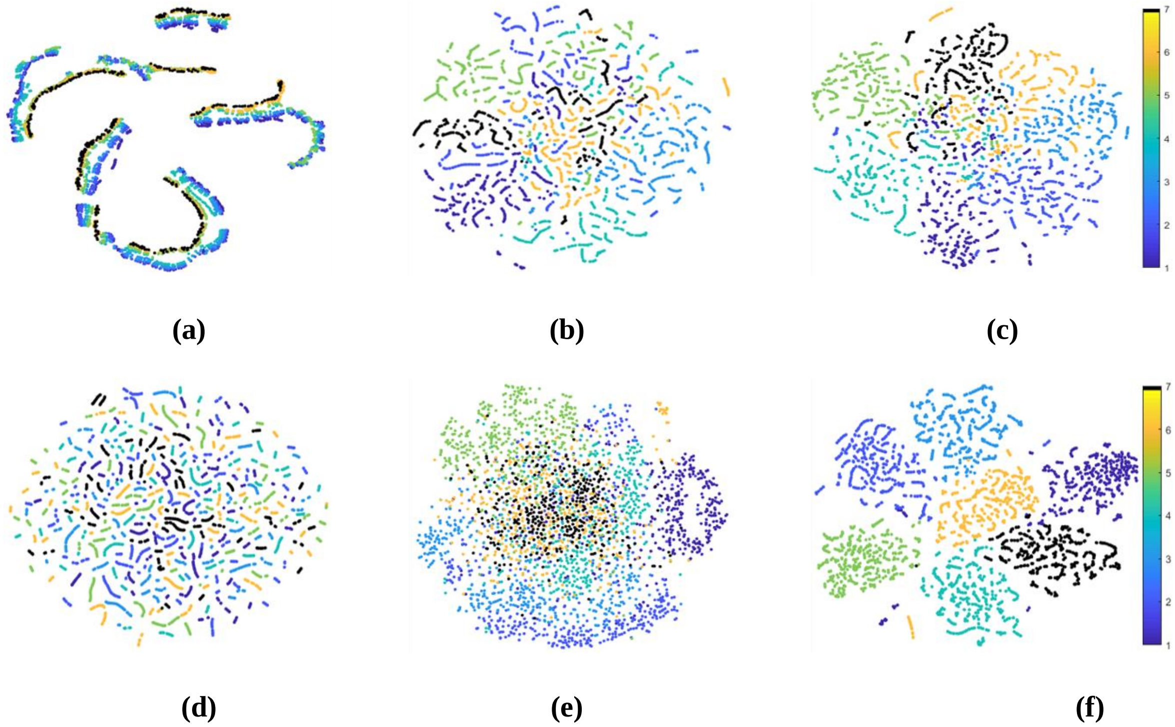

Figure 5.3. Dataset 1 - (Top) Feature Visualization of an unseen participant using t-SNE – UI-M-CNN. (Bottom) Feature Visualization of an unseen participant using t-SNE – UI-C-CNN. (a) Input magnitude spectrum features (Left). (b) Output of Conv_1_ReLU Layer of M-CNN (Middle). (c) Output of the Conv_2_ReLU layer of M-CNN (Right). (d) Input complex spectrum features (Left). (e) Output of Conv_1_ReLU Layer of C-CNN (Middle). (f) Output of the Conv_2_ReLU layer of C-CNN (Right).

complex representation of the inputs. Therefore, seven unique classes corresponding to the seven flicker frequencies have been learned by the CNN and has the ability to cluster the unseen participant's data into these classes. The overlap between clusters is reduced in C-CNN when compared to M-CNN. The classification accuracies confirm these findings for all window lengths where the UI-C-CNN outperforms the UI-M-CNN. Therefore, as a result of combining the real and imaginary parts of the complex FFT, the C-CNN leads to better overall separation and achieves higher classification accuracy when compared to the magnitude spectrum features.

5.5.2 Dataset 2

Figure 5.4 provides a visualization of the data of all participants on Dataset 2, extracted at the output of the Conv_2_ReLU layer for UI-M-CNN and UI-C-CNN. At the output of this layer of UI-C-CNN, clear separation between classes and distinct clusters can be observed compared to UI-M-CNN. This type of class separation aids in achieving better classification accuracy. In [32], the authors provided time domain features, extracted on Dataset 2, as input to a CNN and showed that their proposed CNN was able to capture within-class clusters. In the analysis presented in this thesis, we observed a similar within-class separation learned by the UI-C-CNN when using complex spectrum features which is a lighter CNN architecture. These are illustrated in Figure 5.4. Figure 5.4 (c) illustrates an example of the trials belonging to the 12.25 Hz class. Four distinct clusters can be identified within this class. Subsequent analysis found that these clusters actually correspond to the four non-overlapping 1 s segments

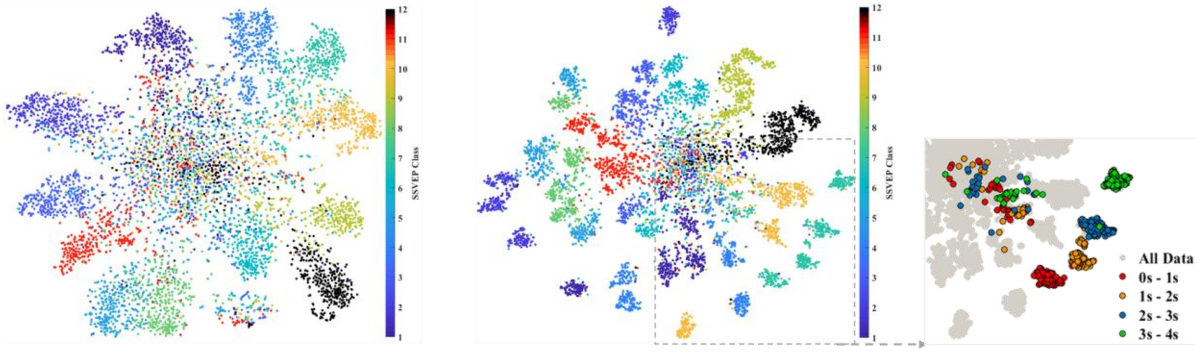

Figure 5.4. Dataset 2 - Feature Visualization of all participants using t-SNE. (a) Output of the Conv_2_ReLU layer of UI-M-CNN (Left). (b) Output of the Conv_2_ReLU layer of UI-C-CNN (Middle). (c) Segment level clustering for SSVEP Class 12.25 Hz of UI-C-CNN (Right).

of the 4 s trials of the class belonging to 12.25 Hz data. These four clusters were colored according to the segment label. As a result, a segment level clustering was observed in the representations learned by the CNN, i.e. all the 1 s segments were clustered into four groups as follows: 0s - 1s, 1s - 2s, 2s - 3s and 3s - 4s. This separation could be due to the existence of phase related information within the segments and the C-CNN has likely learned to extract the phase and amplitude related information directly from the complex representation of the input. These findings are consistent with the results reported in [32]. From these results, it is evident that the proposed UI-C-CNN method can improve the overall SSVEP decoding performance significantly. The five methods that were presented in this thesis were CCA, UD-M-CNN, UD-C-CNN, UI-M-CNN, and UI-C-CNN. The results achieved on Dataset 2 were compared with other studies in the literature that have reported their findings based on this dataset. This was motivated by the recommendation of a recent study [16]. The authors reported that a vast majority of published studies based on deep learning for EEG based BCIs did not compare the proposed techniques to state-of-the-art methods or they performed biased comparisons. Therefore, we have attempted to address this issue by comparing our methods with other techniques proposed in the literature. Therefore we compared two UD and two UI methods as identified in [51] with the five methods used in this thesis. The combination method [22] and Independent Template based CCA (IT-CCA) [21] were selected among the UD methods. The Compact-CNN [32] and the Combined-tCCA [23] methods were selected among the UI methods. All these methods were tested by the respective studies and they reported the results for the 1 s data. Figure 5.5 presents the classification accuracies of the calibration-free CCA, UD and UI training methods presented in this thesis along with the values reported in the literature. Compared to UI methods and CCA, the UD methods achieve a higher performance. The proposed UD-C-CNN (92.3±11.1%) outperforms UD-M-CNN (82.8±16.7%), IT-CCA (81.2±18.84%) and CCA (62.7±21.5%), but has similar performance compared to the Combination method (92.8±10.22). Among the UI methods, the proposed UI-C-CNN

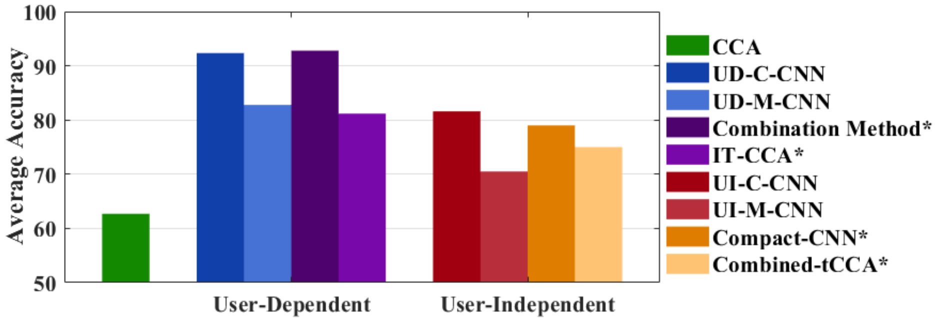

Figure 5.5. Comparing the UD and UI methods on Dataset 2 for 1 s window length with other methods as reported in the literature. *Values used directly from the respective studies.

(81.6±18%) achieved the highest performance compared to UI-M-CNN (70.5±22%), Compact-CNN (79±15%), Combined-tCCA (75±24%) and CCA (62.7±21.5%).

In summary, this analysis investigated the performance of the proposed CNN methods in both the UI and UD training scenarios. The proposed C-CNN method was compared with the magnitude spectrum features (M-CNN) and CCA. Across all *W*, the C-CNN outperformed both M-CNN and CCA in both UI and UD training scenarios. As expected, the UD methods achieved higher performance than the UI methods. The UD-C-CNN and UI-C-CNN ranked highest among each training category respectively. It is interesting to note that the UI-C-CNN performed similarly to the UD-M-CNN. The results on Dataset 2 were consistent with the ones reported other studies. This was confirmed by the feature visualization step which indicated that C-CNN likely learned phase related information from the SSVEP data. Moreover, UI-C-CNN also achieved highest accuracy among tested UI methods on the public dataset performing similarly to the Combined method, which was the best SSVEP decoder in [22].

Chapter 6
Conclusions and Future Work

Three types of analyses were presented in this thesis with the objective of enhancing the decoding performance of an SSVEP BCI. Various system design parameters such as stimulus design, EEG channel selection, detection algorithms and training scenarios were explored. Two main goals were achieved in this thesis: (i) enhance the performance under variable ISD through channel selection and through a novel classification algorithm, the C-CNN (ii) development of a UI training method.

The first analysis revealed that the performance of the user-specific three channel set outperformed the classic 3-channel set and 6-channel sets. Simultaneously, these user-specific channels were shown to be less variable against changing ISD. Next, a novel method, C-CNN was proposed in which the complex spectrum of the SSVEP was provided as input to a CNN. This method outperformed the M-CNN method and CCA classification, and has the ability to generalize across new ISDs and sessions. Both these methods prove to be robust under the influence of competing stimuli. In the former case, it is required to collect calibration data of 20 minutes for selecting the best EEG channels for each user. Whereas in the latter case, calibration data of just 12 minutes is required to calibrate the detection algorithm with fixed set of EEG channels (O1, O2 and Oz). Therefore, the proposed methods achieved an enhanced decoding performance of SSVEP under the influence of competing stimuli and provides an improved flexibility for SSVEP stimulus interface design.

A comparative analysis was carried out between UD and UI training scenarios. This was done to inform whether the cost of calibration would be borne by the user (in case of UD training) or by the developer of the BCI (in case of UI training). From the analysis we observe that there is a trade-off between achieving a high classification accuracy and the cost of collecting calibration data. For the developer, if the priority was to achieve higher performance, then UD methods offer the best accuracy compared to UI and calibration-free methods. But this means each user must undergo a calibration session, and this could possibly lead to poor user-compliance. On the contrary, if the developer of the system invests in collecting calibration data from multiple users, then the UI-C-CNN method proposed in this thesis offers a good balance between performance and cost of calibration.

A combined advantage of high accuracy of the UD training scheme and simultaneously having a low calibration cost can be achieved by using transfer learning based methods. This can be explored in a future study where a model can be developed with multiple participants' data and by collecting minimal calibration data from the unseen user, the pre-trained model can be fine-tuned. Furthermore, online adaptation strategies can be explored for improving the overall performance of the BCI. In this thesis, the UD and UI methods were evaluated on two datasets consisting of 21 participants and 10 participants each. We recommend that a future study explores the number of users required to build a sufficiently accurate UI model. All the analyses presented in this thesis have been performed in an offline manner, therefore subsequent studies must be carried out to validate the methods in an online scenario.

In summary, the proposed methods are suitable candidates for SSVEP-based BCIs. They provide an improved performance under effects of competing stimuli and also in both user-dependent and user-independent training scenarios.

References

[1] A. Ravi, S. Pearce, X. Zhang, N. Jiang, and S. Member, “User-Specific Channel Selection Method to Improve SSVEP BCI Decoding Robustness Against Variable Inter-Stimulus Distance,” *9th Int. IEEE EMBS Conf. Neural Eng.*, pp. 283–286, 2019.

[2] A. Ravi, J. Manuel, N. Heydari, and N. Jiang, “A Convolutional Neural Network for Enhancing the Detection of SSVEP in the Presence of Competing Stimuli,” in *IEEE Engineering in Medicine and Biology,* 2019, p. (Accepted).

[3] A. Ravi, N. Heydari, and N. Jiang, “User-Independent SSVEP BCI Using Complex FFT Features and CNN Classification,” in *2019 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2019,* p. (Accepted).

[4] G. Dornhege, J. R. Millán, T. Hinterberger, D. Mcfarland, K. Müller, and A. B. Book, “Towards Brain-Computer Interfacing,” pp. 31–42, 2007.

[5] B. Z. Allison, D. J. McFarland, G. Schalk, S. D. Zheng, M. M. Jackson, and J. R. Wolpaw, “Towards an independent brain-computer interface using steady state visual evoked potentials,” *Clin. Neurophysiol.*, vol. 119, no. 2, pp. 399–408, 2008.

[6] D. Lesenfants *et al.*, “An independent SSVEP-based brain-computer interface in locked-in syndrome,” *J. Neural Eng.*, vol. 11, no. 3, 2014.

[7] X. Zhang, G. Xu, J. Xie, M. Li, W. Pei, and J. Zhang, “An EEG-driven Lower Limb Rehabilitation Training System for Active and Passive Co-stimulation,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2015–Novem, no. August, pp. 4582–4585, 2015.

[8] L. F. Nicolas-Alonso and J. Gomez-Gil, “Brain computer interfaces, a review,” *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.

[9] J. J. Vidal, “Toward Direct Brain-Computer Communication,” *Annu. Rev. Biophys. Bioeng.*, vol. 2, no. 1, pp. 157–180, 1973.

[10] Y. Jiang, X. Zhao, R. Abiri, S. Borhani, and E. W. Sellers, “A comprehensive review of

EEG-based brain–computer interface paradigms," *J. Neural Eng.*, vol. 16, no. 1, p. 011001, 2018.

[11] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain-computer communication," *Proc. IEEE*, vol. 89, no. 7, pp. 1123–1134, 2001.

[12] F. Karimi, J. Kofman, N. Mrachacz-Kersting, D. Farina, and N. Jiang, "Detection of movement related cortical potentials from EEG using constrained ICA for brain-computer interface applications," *Front. Neurosci.*, 2017.

[13] W. Yan, G. Xu, J. Xie, M. Li, and Z. Dan, "Four novel motion paradigms based on steady-state motion visual evoked potential," in *IEEE Transactions on Biomedical Engineering*, 2018.

[14] S. P. Kelly, E. Lalor, C. Finucane, and R. B. Reilly, "A comparison of covert and overt attention as a control option in a steady-state visual evoked potential-based brain computer interface," pp. 4725–4728, 2005.

[15] C. Guger *et al.*, "How many people could use an SSVEP BCI?," *Front. Neurosci.*, vol. 6, no. NOV, pp. 2–7, 2012.

[16] F. Lotte *et al.*, "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update," *J. Neural Eng.*, pp. 0–20, 2018.

[17] R. V. Sastry, R. Aravind, G. Panwar, S. Indrapriyadarsini, and A. G. Ramakrishnan, "Automatic user customized brain switch," in *2016 IEEE Annual India Conference, INDICON 2016*, 2017.

[18] Zhonglin Lin, Changshui Zhang, Wei Wu, and Xiaorong Gao, "Frequency Recognition Based on Canonical Correlation Analysis for SSVEP-Based BCIs," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2610–2614, 2006.

[19] G. Bin, X. Gao, Z. Yan, B. Hong, and S. Gao, "An online multi-channel SSVEP-based

brain-computer interface using a canonical correlation analysis method," *J. Neural Eng.*, 2009.

[20] X. Chen, Y. Wang, S. Gao, T. P. Jung, and X. Gao, "Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain-computer interface," *J. Neural Eng.*, vol. 12, no. 4, 2015.

[21] Y. Wang, M. Nakanishi, Y. Te Wang, and T. P. Jung, "Enhancing detection of steady state visual evoked potentials using individual training data," *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014*, pp. 3037–3040, 2014.

[22] M. Nakanishi, Y. Wang, Y. Te Wang, and T. P. Jung, "A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials," *PLoS One*, vol. 10, no. 10, pp. 1–18, 2015.

[23] N. R. Waytowich, J. Faller, J. O. Garcia, J. M. Vettel, and P. Sajda, "Unsupervised adaptive transfer learning for Steady-State Visual Evoked Potential brain-computer interfaces," in *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, 2017.

[24] M. Nakanishi, Y. Wang, X. Chen, Y. Te Wang, X. Gao, and T. P. Jung, "Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis," *IEEE Trans. Biomed. Eng.*, 2018.

[25] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: a review," *Comput. Methods Programs Biomed.*, 2018.

[26] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: a systematic review," 2019.

[27] A. Craik, Y. He, and J. L. (Pepe) Contreras-Vidal, "Deep learning for

Electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, 2019.

[28] I. G. and Y. B. and A. Courville, *Deep Learning*. 2016.

[29] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, p. 056013, 2018.

[30] N. S. Kwak, K. R. Müller, and S. W. Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PLoS One*, 2017.

[31] T.-H. Nguyen and W.-Y. Chung, "A Single-Channel SSVEP-Based BCI Speller using Deep Learning," *IEEE Access*, vol. 7, pp. 1–1, 2018.

[32] N. Waytowich *et al.*, "Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials," *J. Neural Eng.*, 2018.

[33] X. Zhang *et al.*, "A Convolutional Neural Network for the Detection of Asynchronous Steady State Motion Visual Evoked Potential," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. PP, no. c, pp. 1–1, 2019.

[34] H. J. Hwang, K. Kwon, and C. H. Im, "Neurofeedback-based motor imagery training for brain-computer interface (BCI)," *J. Neurosci. Methods*, vol. 179, no. 1, pp. 150–156, 2009.

[35] L. A. and R. F.-R. Setare Amiri, Ahmed Rabbi, "A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain- Computer Interface Systems," in *Brain-Computer Interface Systems - Recent Progress and Future Prospects*, 2013, pp. 195–214.

[36] N. A. *et al.*, "Brain-computer interfacing for intelligent systems," *IEEE Intell. Syst.*, vol. 23, pp. 72–79, 2008.

[37] C. Vidaurre and B. Blankertz, "Towards a cure for BCI illiteracy," *Brain Topogr.*, vol.

23, no. 2, pp. 194–198, 2010.

[38] I. Volosyak, D. Valbuena, T. Lüth, T. Malechka, and A. Gräser, “BCI demographics II: How many (and What Kinds of) people can use a high-frequency SSVEP BCI?,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 3, pp. 232–239, 2011.

[39] K. B. Ng, A. P. Bradley, and R. Cunnington, “Stimulus specificity of a steady-state visual-evoked potential-based brain-computer interface,” *J. Neural Eng.*, 2012.

[40] D. Regan, “Some Characteristics of Average Steady-State,” *Electroencephalogr. Clin. Neurophysiol.*, pp. 238–248, 1965.

[41] W. Yijun, W. Ruiping, G. Xiaorong, and G. Shangkai, “Brain-computer Interface based on the High-frequency Steady-state Visual Evoked Potential,” vol. 26, 2005.

[42] Y. Wang, R. Wang, X. Gao, B. Hong, and S. Gao, “A practical VEP-based brain-computer interface,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 234–239, 2006.

[43] T. Cao, F. Wan, P. U. Mak, P. I. Mak, M. I. Vai, and Y. Hu, “Flashing color on the performance of SSVEP-based brain-computer interfaces,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2012.

[44] A. Duszyk *et al.*, “Towards an optimization of stimulus parameters for brain-computer interfaces based on steady state visual evoked potentials,” *PLoS One*, vol. 9, no. 11, pp. 1–11, 2014.

[45] K. B. Ng, A. P. Bradley, and R. Cunnington, “Effect of competing stimuli on SSVEP-based BCI,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2011.

[46] F. B. Vialatte, M. Maurice, J. Dauwels, and A. Cichocki, “Steady-state visually evoked

potentials: Focus on essential paradigms and future perspectives," *Progress in Neurobiology*. 2010.

[47] D. Zhu, J. Bieger, G. Garcia Molina, and R. M. Aarts, "A survey of stimulation methods used in SSVEP-based BCIs," *Computational Intelligence and Neuroscience*. 2010.

[48] J. Faller *et al.*, "A feasibility study on SSVEP-based interaction with motivating and immersive virtual and augmented reality," pp. 1–6, 2017.

[49] M. Nakanishi, Y. Wang, Y. Te Wang, Y. Mitsukura, and T. P. Jung, "Generating visual flickers for eliciting robust steady-state visual evoked potentials at flexible frequencies using monitor refresh rate," *PLoS One*, 2014.

[50] G. R. Müller-Putz, E. Eder, S. C. Wriessnegger, and G. Pfurtscheller, "Comparison of DFT and lock-in amplifier features and search for optimal electrode positions in SSVEP-based BCI," *J. Neurosci. Methods*, vol. 168, no. 1, pp. 174–181, 2008.

[51] R. Zerafa, T. Camilleri, O. Falzon, and K. P. Camilleri, "To train or not to train? A survey on training of feature extraction methods for SSVEP-based BCIs," *J. Neural Eng.*, vol. 15, no. 5, 2018.

[52] G. R. Müller-Putz and G. Pfurtscheller, "Control of an electrical prosthesis with an SSVEP-based BCI," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 361–364, 2008.

[53] Y. Zhang, J. Jin, X. Qing, B. Wang, and X. Wang, "LASSO based stimulus frequency recognition model for SSVEP BCIs," *Biomed. Signal Process. Control*, vol. 7, no. 2, pp. 104–111, 2012.

[54] H. Cecotti, "A time-frequency convolutional neural network for the offline classification of steady-state visual evoked potential responses," *Pattern Recognit. Lett.*, vol. 32, no. 8, pp. 1145–1153, 2011.

[55] N. K. Nik Aznan, S. Bonner, J. Connolly, N. Al Moubayed, and T. Breckon, "On the

Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks," *Proc. - 2018 IEEE Int. Conf. Syst. Man, Cybern. SMC 2018*, pp. 3726–3731, 2019.

[56] T. Kluge and M. Hartmann, "Phase coherent detection of steady-state evoked potentials: Experimental results and application to brain-computer interfaces," *Proc. 3rd Int. IEEE EMBS Conf. Neural Eng.*, pp. 425–429, 2007.

[57] J. Pan, X. Gao, F. Duan, Z. Yan, and S. Gao, "Enhancing the classification accuracy of steady-state visual evoked potential-based brain-computer interfaces using phase constrained canonical correlation analysis," *J. Neural Eng.*, vol. 8, no. 3, 2011.

[58] R. D. and J. Duncan, "Neural Mechanisms of Selective Visual Attention," *Annu. Rev. Neurosci.*, pp. 193–222, 1995.

[59] L. Angrisani, P. Arpaia, N. Moccaldi, and A. Esposito, "Wearable Augmented Reality and Brain Computer Interface to Improve Human-Robot Interactions in Smart Industry: A Feasibility Study for SSVEP Signals," *IEEE 4th Int. Forum Res. Technol. Soc. Ind. RTSI 2018 - Proc.*, pp. 1–5, 2018.

[60] X. Gao, D. Xu, M. Cheng, and S. Gao, "A BCI-based environmental controller for the motion-disabled," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2003.

[61] Y. Wang, Y.-T. Wang, and T.-P. Jung, "Visual stimulus design for high-rate SSVEP BCI," *Electron. Lett.*, vol. 46, no. 15, p. 1057, 2010.

[62] Y. Renard *et al.*, "OpenViBE : An Open-Source Software Platform to Design , Test , and Use Brain – Computer Interfaces in Real and Virtual Environments," *Presence*, vol. 19, no. 1, pp. 35–53, 2010.

[63] D. R. Hardoon, S. Szedmak, O. Szedmak, and J. Shawe-taylor, "Canonical correlation analysis; An overview with application to learning methods," *MIT Press Journals - Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2007.

[64] P. Yuan, X. Chen, Y. Wang, X. Gao, and S. Gao, "Enhancing performances of SSVEP-based brain-computer interfaces via exploiting inter-subject information," *J. Neural Eng.*, vol. 12, no. 4, 2015.

[65] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

[66] G. H. Laurens van der Maaten, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[67] D. López-sánchez, A. G. Arrieta, and J. M. Corchado, "Deep neural networks and transfer learning applied to multimedia web mining Deep neural networks and transfer learning applied to multimedia web mining," no. February, 2018.

[68] B. C. Oxley, "International 10-20 system for EEG-MCN." [Online]. Available: https://commons.wikimedia.org/wiki/File:International_10-20_system_for_EEG-MCN.svg (Accessed: 04-July-2019)

Appendix A

MATLAB Scripts

A.1 Convolutional Neural Network MATLAB Implementation

```
%Convolutional Neural Network Definition

%Input Layer
layers = [imageInputLayer([size(Train_data,1) size(Train_data,2) 1])
%Conv_1 Layer
            convolution2dLayer([Nch,1],2*Nch)
      batchNormalizationLayer
      reluLayer
      dropoutLayer(0.25)
%Conv_2 Layer
      convolution2dLayer([1,10],2*Nch)
      batchNormalizationLayer
      reluLayer
      dropoutLayer(0.25)
%Output Layer
      fullyConnectedLayer(K)
      softmaxLayer
      classificationLayer];

% CNN Training Parameters
      options = trainingOptions('sgdm', ...
        'MaxEpochs',40,...
        'MiniBatchSize', 256,...
        'InitialLearnRate',0.001, ...
        'Verbose',1, ...
        'VerboseFrequency',10, ...
        'ValidationFrequency',10, ...
        'Plots','none','ExecutionEnvironment','cpu');

% Train the Network
       net = trainNetwork(train_data, train_lab, layers, options);
```

A.2 User-Specific Channel Selection Method Implementation

```
%User Specific Channel Selection Method
clc;
clear all;
close all;

%Iterating through S1, S2 and S3
for configData = 1:3

%Loading the gdf file using Biosig Toolbox
[data, metaInfo] = sload(strcat('Data/P1_Data/P1_',num2str(configData),'*'));
metaInfo.Label = extractBefore(metaInfo.Label,' ');
%Flicker frequencies of the Stimuli
freqBands = [8.423,9.375,9.961,10.84,11.87,13.4,14.87];

%Enter the number of channel combinations
numChanCombo = 3;
sampleRate = metaInfo.SampleRate;
numChannels = metaInfo.NS;
data = data(:, 1:numChannels); % selection of channels

%Bandpass filtering between 1Hz and 40Hz
order = 4;
lowFreq = 1* (2/sampleRate);
highFreq = 40 * (2/sampleRate);
[B, A] = butter(order, [lowFreq, highFreq]);
signal = filter(B, A, data);

%Epoch range to compute the CCA results
epochRange = [0.5,3.5];
%Time indexes for plotting (samples)
trialTimeIdx = (round(sampleRate*epochRange(1))+1):round(sampleRate*epochRange(2));

stimCodes = [33025, 33026, 33027, 33028, 33029, 33030, 33024];
numClasses = size(stimCodes, 2);

class1StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(1)))+1);
class2StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(2)))+1);
```

```
class3StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(3)))+1);
class4StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(4)))+1);
class5StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(5)))+1);
class6StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(6)))+1);
class7StartTime = metaInfo.EVENT.POS(find(ismember(metaInfo.EVENT.TYP, stimCodes(7)))+1);

for epochID=1:length(class1StartTime)
    class1Epoches(:,:,epochID) = signal(class1StartTime(epochID)+trialTimeIdx,:,:);
end
for epochID=1:length(class2StartTime)
    class2Epoches(:,:,epochID) = signal(class2StartTime(epochID)+trialTimeIdx,:,:);
end
for epochID=1:length(class3StartTime)
    class3Epoches(:,:,epochID) = signal(class3StartTime(epochID)+trialTimeIdx,:,:);
end
for epochID=1:length(class4StartTime)
    class4Epoches(:,:,epochID) = signal(class4StartTime(epochID)+trialTimeIdx,:,:);
end
for epochID=1:length(class5StartTime)
    class5Epoches(:,:,epochID) = signal(class5StartTime(epochID)+trialTimeIdx,:,:);
end
for epochID=1:length(class6StartTime)
    class6Epoches(:,:,epochID) = signal(class6StartTime(epochID)+trialTimeIdx,:,:);
end
for epochID=1:length(class7StartTime)
    class7Epoches(:,:,epochID) = signal(class7StartTime(epochID)+trialTimeIdx,:,:);
end
%PO3 POz PO4 O1 Oz O2
channelsCombo = combnk([6 7 8 11 12 13],numChanCombo);
accuracy=[];

for indexI=1:length(channelsCombo)
               allTrials = cat(3,class1Epoches,class2Epoches,class3Epoches,class4Epoches,class5Epoches,class6Epoches,class7Epoches);
               labels =
[ones(1,length(class1StartTime)),2*ones(1,length(class2StartTime)),3*ones(1,length(class3StartTime)),4*ones(1,length(class4StartTime)),
5*ones(1,length(class5StartTime)),6*ones(1,length(class6StartTime)),7*ones(1,length(class7StartTime))];
        for epochID=1:size(allTrials,3)
                  %Compute CCA Coefficients
       r_coeff=computeCCA(allTrials(:,channelsCombo(indexI,:),epochID),sampleRate,freqBands);
            [R ind] = sort(r_coeff,'descend');
            result(epochID,1)=ind(1);
        end
        Cmat = confusionmat(result,labels);
        accuracy = [accuracy,trace(Cmat)/(sum(sum(Cmat)))];

end
accuracyFinal(:,configData) = accuracy';
end

%Channel Selection Algorithm
meanAcc = mean(accuracyFinal,2);
variationAcc = max(accuracyFinal,[],2)-min(accuracyFinal,[],2);
variationAcc = round(variationAcc,4);

%Sort the accuracies in descending order
[sortedAcc chanAccInd] = sort(meanAcc,'descend');

%Pick the top 20% based on accuracies
percToSelect = round(0.2*length(chanAccInd));
topFour = chanAccInd(1:percToSelect,:);

%Pick the top four variances
topFourVar = variationAcc(topFour,:);
topFourAcc = sortedAcc(topFour);

%Sort in ascending order the variances
[sortedVar leastChangeInd] = sort(topFourVar);

%sortedVar(1,1) contains the least variation
varOccurence = histc(sortedVar,sortedVar(1,1));

if varOccurence > 1
    multipleIdx = find(sortedVar(1,1)==topFourVar);
    preSelection = find(max(sortedAcc(multipleIdx))==sortedAcc(multipleIdx));
    tempIdx = chanAccInd(multipleIdx);
    selecChannel = tempIdx(preSelection);
else
    multipleIdx = find(sortedVar(1,1)==topFourVar);
    selecChannel = chanAccInd(multipleIdx);
end
```

```
%Display the selected User-Specific Channel Set and corresponding Accuracies
bestChan = channelsCombo(selecChannel,:);
finalChanSelection = [metaInfo.Label{bestChan(1,1)},metaInfo.Label{bestChan(1,2)},metaInfo.Label{bestChan(1,3)}]
finalAcc = accuracyFinal(selecChannel,:)
```

A.3 Canonical Correlation Analysis (CCA) MATLAB Implementation

```
% Canonical Correlation Analysis (CCA)
function rmax=computeCCA(data,sampleRate,stimFreq)
dataLen = length(data);
timeAxis = (1/sampleRate:1/sampleRate:dataLen/sampleRate)';
numberTargets=length(stimFreq);

%Collect the values
rmax=zeros(1,numberTargets);

for classNum=1:numberTargets
    freq=2*pi*stimFreq(classNum)*timeAxis;
    %Harmonics
    freqSet=[freq,freq*2];
    cosRef=cos(freqSet);
    sinRef=sin(freqSet);
    %Apply CCA
    [Wx,Wy,r1,U,V] = canoncorr(data,[cosRef,sinRef]);
    rmax(classNum) = rmax(classNum)+max(r1);
end
```

Appendix B

Individual Participants Results

B.1 Study I - User-Specific Channel Selection Results

Table B.1 User-Specific channel selection accuracies (%) for individual participants. Improvements in accuracy for UC set compared to 3C or 6C set highlighted in grey

	3C-Set			**6C-Set**			**UC-Set**		
	S_1	**S_2**	**S_3**	**S_1**	**S_2**	**S_3**	**S_1**	**S_2**	**S_3**
S01	94.64	98.21	98.21	96.43	98.21	94.64	94.64	98.21	98.21
S02	91.07	96.43	100.00	96.43	96.43	100.00	96.43	96.43	96.43
S03	71.43	91.07	87.50	87.50	91.07	89.29	91.07	92.86	92.86
S04	91.07	100.00	98.21	94.64	100.00	98.21	98.21	100.00	98.21
S05	85.71	89.29	96.43	83.93	89.29	91.07	94.64	96.43	96.43
S06	85.71	100.00	100.00	92.86	100.00	100.00	94.64	98.21	100.00
S07	98.21	98.21	96.43	96.43	96.43	94.64	98.21	98.21	98.21
S08	80.36	91.07	89.29	91.07	87.50	89.29	94.64	91.07	91.07
S09	83.93	98.21	98.21	85.71	92.86	92.86	87.50	92.86	94.64
S10	98.21	98.21	98.21	94.64	100.00	100.00	98.21	100.00	100.00
S11	85.71	92.86	92.86	94.64	82.14	89.29	85.71	92.86	92.86
S12	83.93	94.64	92.86	89.29	98.21	96.43	96.43	96.43	100.00
S13	80.36	78.57	75.00	75.00	66.07	73.21	80.36	83.93	78.57
S14	94.64	94.64	96.43	100.00	98.21	100.00	100.00	98.21	100.00
S15	87.50	87.50	92.86	87.50	87.50	94.64	92.86	91.07	94.64
S16	94.64	100.00	98.21	100.00	100.00	98.21	100.00	100.00	100.00
S17	83.93	82.14	92.86	82.14	83.93	92.86	89.29	89.29	92.86
S18	78.57	94.64	92.86	96.43	96.43	98.21	94.64	94.64	92.86
S19	78.57	96.43	100.00	82.14	96.43	100.00	82.14	96.43	100.00
S20	92.86	98.21	96.43	100.00	98.21	98.21	100.00	98.21	98.21
S21	66.07	82.14	80.36	67.86	87.50	91.07	69.64	80.36	83.93
Average	**86.05**	**93.45**	**93.96**	**90.22**	**92.69**	**94.39**	**92.35**	**94.56**	**95.24**

B.2 Study II - Results of CNN Robust to Change in ISD

Table B.2 Comparing the accuracies (%) for individual participants of CCA, M-CNN and C-CNN across different ISDs for 1 s window length (O1-O2-Oz).

	CCA			**M-CNN**			**C-CNN**		
	S_1	**S_2**	**S_3**	**S_1**	**S_2**	**S_3**	**S_1**	**S_2**	**S_3**
S01	66.96	82.89	80.28	88.08	97.08	95.61	89.61	96.94	96.76
S02	68.30	80.92	83.37	65.60	76.77	72.23	82.38	92.35	92.00
S03	62.87	76.38	77.01	76.35	83.58	80.34	91.71	96.23	91.67
S04	83.44	93.08	91.56	95.74	99.36	98.71	96.81	99.45	99.18
S05	62.09	68.56	69.35	82.78	86.52	87.10	86.32	90.03	90.03
S06	68.08	88.95	91.22	87.90	98.27	98.72	85.70	97.38	98.47
S07	80.69	84.49	88.28	87.26	89.25	91.62	91.51	93.09	96.05
S08	68.42	71.17	76.38	82.43	81.47	86.04	86.35	84.46	87.37
S09	58.85	74.78	72.32	50.87	57.89	61.22	74.33	87.48	86.24
S10	84.15	87.57	92.86	94.30	96.48	98.87	97.50	98.51	98.89
S11	61.42	62.46	67.97	81.10	71.74	77.77	91.67	85.17	90.62
S12	69.68	79.69	81.25	71.99	77.30	81.87	89.47	93.44	95.99
S13	57.18	50.89	54.32	68.11	61.95	69.55	75.98	64.18	72.65
S14	81.58	76.75	76.38	93.70	89.83	89.30	96.45	93.26	92.62
S15	75.04	74.59	78.79	77.81	79.14	82.31	89.32	90.62	93.42
S16	85.57	94.05	88.36	94.11	99.00	98.47	95.70	99.14	98.78
S17	74.70	77.27	76.23	80.56	79.76	80.43	88.47	92.71	90.63
S18	59.08	73.29	70.94	55.47	68.15	64.43	78.10	90.03	84.77
S19	58.93	80.84	81.66	59.60	77.72	78.02	67.18	85.20	85.11
S20	80.77	86.01	87.28	88.78	92.00	92.42	89.12	93.09	93.84
S21	43.45	66.00	55.88	63.96	84.64	79.94	63.68	84.82	79.30
Average	**69.11**	**77.65**	**78.17**	**78.40**	**83.23**	**84.05**	**86.06**	**90.84**	**91.16**

B.3 Study III - Dataset 1 – UD vs. UI Comparison

Table B.3 Dataset 1 - Individual accuracies (%) comparing UD and UI methods of 1 s window length and ISD S_1 (O1-O2-Oz)

	CCA	**UI-M-CNN**	**UI-C-CNN**	**UD-M-CNN**	**UD-C-CNN**
S01	66.96	77.62	81.60	94.20	95.08
S02	68.30	59.62	79.26	81.60	90.20
S03	62.87	71.47	89.32	86.33	95.95
S04	83.44	95.19	94.21	97.96	98.61
S05	62.09	78.86	85.35	88.88	91.04
S06	68.08	89.40	89.47	94.46	94.17
S07	80.69	85.90	90.20	92.17	96.03
S08	68.42	84.29	87.57	89.51	91.00
S09	58.85	43.19	63.12	73.03	86.30
S10	84.15	94.21	97.56	96.68	99.02
S11	61.42	73.32	85.20	89.91	94.39
S12	69.68	65.09	85.35	85.35	94.61
S13	57.18	67.13	72.05	83.20	86.41
S14	81.58	88.78	93.40	97.08	97.63
S15	75.04	69.06	75.62	85.97	92.78
S16	85.57	95.59	94.90	96.94	98.36
S17	74.70	73.76	88.08	86.44	94.79
S18	59.08	46.98	64.36	71.10	87.39
S19	58.93	55.39	61.19	76.97	81.38
S20	80.77	84.66	86.04	95.26	95.59
S21	43.45	44.31	50.66	81.71	82.18
Average	**69.11**	**73.51**	**81.64**	**87.85**	**92.52**

B.4 Study III - Dataset 2 – UD vs. UI Comparison

Table B.4 Dataset 2 - Individual accuracies (%) comparing UD and UI methods of 1 s window length (8 Channels in dataset)

	CCA	UI-M-CNN	UI-C-CNN	UD-M-CNN	UD-C-CNN
S01	27.64	36.67	61.39	65.56	80.42
S02	27.22	20.69	36.53	40.42	63.19
S03	58.75	67.50	76.94	82.92	96.81
S04	79.72	88.75	94.58	91.25	98.61
S05	51.81	83.75	87.64	94.58	99.72
S06	86.25	86.11	93.89	95.83	99.72
S07	66.39	76.67	87.36	89.31	95.42
S08	96.11	96.25	97.64	97.78	99.44
S09	67.78	74.17	93.33	90.00	97.92
S10	65.28	75.14	86.39	80.14	92.08
Average	**62.69**	**70.57**	**81.57**	**82.78**	**92.33**