Paradigm-Independent Classification on Multidimensional Neuroimaging Dataset Using Convolutional Neural Networks

by

Kanat Alimanov

B.S., Nazarbayev University (2019)

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at the

NAZARBAYEV UNIVERSITY

May 2021

Author .....
Department of Computer Science
May 3, 2021

Certified by .....
Min-Ho Lee
Assistant Professor
Thesis Supervisor

Certified by .....
Kok-Seng Wong
Associate Professor
Thesis Co-Supervisor

Accepted by .....
Vassilios D. Tourassis
Dean, School of Engineering and Digital Sciences

Paradigm-Independent Classification on Multidimensional Neuroimaging Dataset Using Convolutional Neural Networks

by

Kanat Alimanov

Submitted to the Department of Computer Science
on May 3, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

Brain-Computer Interfaces (BCI) are gaining popularity with each day. A lot of research was carried out to improve both accuracy and usability of individual BCI paradigms, most popular in current literature being Motor Imagery(MI), Event-Related Potential (ERP)and Steady-State Visually Evoked Potential (SSVEP). However, individual BCI paradigms are limited in some areas such as number of classes in cases of MI and SSVEP paradigms, illiteracy in case of MI, or fatigue induction in users in case of ERP amd SSVEP paradigm. This study presents a new approach to designing BCI systems called paradigm-independent BCI, which solves these problems by allowing the subjects to use any of the three paradigms at any time. For each paradigm, the system processes EEG signals and feeds them through a deep learning model to get a representation vector, these vectors are then concatenated and fed to a final classifier that is used to decide which paradigm is currently being used on the fly. Average classification accuracy for three-class between-paradigm decoding was 97.51%($\pm$3.39) in subject-independent training. Classification accuracy for seven-class within-paradigm decoding resulted in 87.9%($\pm$4.51) accuracy. The results show that this framework works equally great in both subject-dependent and subject-independent contexts.

Thesis Supervisor: Min-Ho Lee
Title: Assistant Professor

Thesis Co-Supervisor: Kok-Seng Wong
Title: Associate Professor

Acknowledgments

I want to thank the reader for their patience.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Brain-Computer Interfaces(BCIs) allow subjects to interact with digital devices by reading the user's brain activity and interpreting it as one of the pre-defined set of commands. Its widespread use can help disabled people gain an easy way of interfacing with a wide array of devices previously unavailable to them. Typical BCI pipeline involves recording subject's brain activity, extracting relevant features from the resulting signal, and running a classification or regression algorithm to calculate the subject's intent. BCI systems achieve this by employing so-called paradigms. Paradigms work by either relying on the users to invoke the correct signals themselves or eliciting meaningful signals in user's brain by exposing them to specific stimuli and measuring their response. There are three main BCI paradigms: Motor Imagery (MI), Event-Related Potential (ERP), and Steady-State Visually Evoked Potential (SSVEP). In the MI paradigm, users are asked to imagine moving one of their appendages. This imagined movement elicits an Event-Related Desynchronization (ERD) in brain's sensorimotor cortex. By measuring the brain activity in this sensorimotor cortex area, we can then determine which appendage's movement the user has imagined [1, 2]. Event-Related Potential (ERP) and Steady-State Visually Evoked Potential (SSVEP) paradigms, alternatively, rely on visual stimuli to elicit the response in user's brain. In the case of ERP, as a reaction to an event, voltage deflections can be observed in the subject's brain signals. These deflections differ in their amplitude and occurrence time after stimulus. There are five main ERPs: P1, N1, P2, N2, P3. Usually, to boost

the ERP amplitudes, an oddball paradigm is used, in which users are shown different sequences of stimuli, where the target stimuli appears less frequently than non-target ones, and if the target stimuli is present in this sequence, a corresponding event-related potential can be measured in user's brain shortly after [3, 4]. One common use of this paradigm is in so-called ERP spellers, where different sequences of letters are lit up to determine the letter that the user is focused on. SSVEP, on the other hand relies on visual stimuli that flash at a certain frequency. If the user's gaze is focused on this flashing stimuli, its frequency, along with its harmonics can be measured in the user's visual cortex [5].

Since the obtained EEG signals from the individual paradigms contain distinct neural responses, each of these systems requires its own unique processing pipeline. For instance, the event-related potentials are primarily evoked in the central and visual cortexes, and main ERP components such as P300 or N200 are observed in intervals between 50 to 800 ms with respect to stimulus onset. To estimate discriminant temporal intervals and also to enhance the signal-to-noise ratio (SNR), various preprocessing steps such as trial averaging, artifact rejection, and spatio-temporal filtering methods [6] are commonly applied for ERP-based applications. In the MI paradigm, as a result of motor imagery, neurons that usually fire together desynchronize, producing an ERD in user-specific frequency bands (e.g μ- and β-bands) in the motor cortex. Advanced signal processing and data-driven optimization algorithms have been proposed to enhance the performance of the MI paradigm (e.g. CSP [7], FBCSP [8], CSSP [9], and others). Among the reasons that necessitate the use of these algorithms are the difficulty of performing the imagery tasks and user's unfamiliarity with them(illiteracy) as well as a big brain pattern diversity across individual users. For these reasons the average decoding accuracy across subjects has been the lowest for the MI paradigm when compared to the other two paradigms [10]. In the SSVEP paradigm, advanced filtering algorithms such as CCA [11] are commonly applied to extract and analyze the frequency components of the EEG signal. All of these approaches are industry standard and have shown reliable performance.

While these three aforementioned paradigms are undoubtedly useful, each of them

has their own setbacks such as high illiteracy rate and low amount of available classes for MI, high classification time and eye strain for ERP, excessive eye strain for SSVEP, etc. One possible way of solving these problems is to combine multiple paradigms. This research proposes a BCI framework that unifies these paradigms, called paradigm-independent BCI. This approach will allow users to overcome the drawbacks of each individual BCI paradigm, by allowing users to switch from one paradigm to the other without explicitly notifying the system itself. Given the EEG signal, the system can determine which paradigm the user is currently participating in and run classification accordingly. This process is handled by CNN networks that, given a processed EEG signal, generates a feature vector that allows us to determine what paradigm and which class in particular is currently being used.

Prior work on paradigm-independent BCI was investigated in [12]. There, however, each paradigm was treated just as one class with the use of RLDA to determine which paradigm is currently in use based on the pre-processed EEG signals. The study showed that paradigm selection can be achieved using a linear RLDA model with reliable 74.84% accuracy. It also highlighted pairwise between-paradigm classification accuracies among SSVEP, MI and ERP paradigms, revealing that discerning between MI and SSVEP paradigms is harder than between MI and ERP or ERP and SSVEP [12].

Chapter two of this work describes the previous attempts at combining multiple BCI paradigms and introduces three approaches to designing multi-paradigm systems. It also touches on the shortcomings of the previous approaches and introduces the paradigm-independent framework that is designed to solve these shortcomings.

Chapter three describes the dataset chosen for this study and presents reasons for choosing it.

Chapter four describes pre-processing steps that were undertaken to extract meaningful matrices for each paradigm present in the framework.

Chapter five details the framework's architecture and explains the steps that are taken to reach the final classification results. It breaks down the architecture of each individual sub-model in the framework.

Chapter six details the performance evaluation steps and presents the experimental results for both subject-dependent and subject-independent cases.

Finally, in chapter seven, the potential of this framework is discussed in detail, with additional suggestions for the future research.

Chapter 2

BCI paradigm combination techniques

Unimodal brain-computer interfaces each have their own drawbacks. Since there is a limited amount of appendages, there is a limited amount of available classes in the MI paradigm. Similar drawback can be observed in the SSVEP paradigm, where the number of classes is highly dependent on the type of monitor that is used to present the flashing stimuli, higher refresh rate allows to present more non-overlapping frequencies, which leads to more classes. ERP and SSVEP paradigms also require constant gaze and attention, which induces eye strain and fatigue in users. These setbacks can potentially be solved by combining multiple paradigms. Common approaches of combining BCI paradigms: Paradigm Incorporation, Paradigm Extension, and Paradigm Fusion are presented next in this section.

2.1 Paradigm Incorporation

First BCI combination technique is Paradigm Incorporation, in which one paradigm becomes a sub-component of another paradigm. This technique can be applied to paradigms that rely on the same type of stimuli. In practice it usually consists of combining ERP and SSVEP paradigms. This approach is employed by Chang et al. [13]. Their paper proposes the use of a hybrid speller that works by flashing nine panels each

at differing frequencies (SSVEP component), each panel also contains four letters that in turn flash at their own frequency that is sub-harmonic to the parent panels frequency. This results in each panel having its own tiny ERP speller inside(ERP component). This approach allows the system to more precisely locate the target ERP stimuli by measuring the SSVEP frequency in the user's visual cortex. Authors claim that use of this hybrid paradigm permits addition of more classes than in classical SSVEP, since it allows for reuse of harmonic frequencies. This hybrid paradigm resulted in a faster information transfer rate (31.8 bpm) than conventional SSVEP based (13 bpm) and P300 based (19.9 bpm) spellers, all with a higher average accuracy [13].

2.2 Paradigm Extension

Second paradigm combination technique is paradigm extension, one defining feature of this technique is that each participating paradigm results in a different set of classification outputs, increasing the final number of classes that is available to the system. This hybridization approach was used by both Duan et al. [14] and Wang et al. [15]. In the study by Duan et al., SSVEP paradigm was combined with MI paradigm to create a system for controlling robot's movement. SSVEP paradigm was used to control robot's movement using three classes: left, right, and forward. MI paradigm with one class (foot motor imagery) was introduced on top of that to add grasping motion to the robot [14]. Two paradigms were used to classify two separate outputs, robot movement and robot arm control. The paper by Wang et al. also combines MI and SSVEP paradigms, but this time to control the game of Tetris. Left hand MI and right hand MI were used to move pieces left and right correspondingly; an SSVEP stimulus was used to rotate the tetromino piece [15]. Both resulting paradigms use additional fusion techniques to determine the output class so that there is only one classification result derived from the original paradigms. Wang et al. uses priority-based hierarchical fusion, giving more priority to the rotation of the piece than to moving it left or right. If two commands from two classifiers are detected at the same time, the one with the higher priority gets executed [15]. Duan et al., on

the other hand, measure brain's μ-rhythm amplitude, and switch the paradigm if this amplitude exceeded a specific threshold, the μ-rhythm was modulated by closing the subject's eyes [14].

2.3 Paradigm Fusion

Third and final BCI combination technique is paradigm fusion, which unlike paradigm extension does not result in new classification classes, and unlike paradigm incorporation does not rely on the other paradigms to produce an output. Usually one paradigm is selected as the main one, and other paradigms are added onto that to reinforce the results. This approach is used by Leeb et al. [16] and Li et al. [17]. Leeb et al. improves the results of right/left hand electromyography (EMG) by adding motor imagery signals to reinforce the output of the system. EMG are the electrical signals that are recorded from the subject's muscle contractions. The signals were fused using what authors call Bayesian fusion, which uses normalized confusion matrices to calculate probabilities of each class occurring using Bayes rule. The authors attenuated EMG signals to simulate subject's fatigue. Results showed that using MI along with EMG data improved the accuracy by 8% and helped alleviate some effects of fatigue [16]. In the work by Li et al. authors combine SSVEP and ERP paradigms to create a system for controlling a wheelchair. The GUI of proposed system consists of four groups of buttons, each group changing color at a certain frequency, which elicits an SSVEP response. Large buttons in the middle of each group are randomly selected, and their color and shape is changed to evoke an event-related potential. To perform the classification, the EEG data is ran through two parallel pipelines (SSVEP classification and ERP classification), combining outputs of both pipelines for the final decision of selecting the button group. SVM classifier was used for the ERP part of the pipeline. SSVEP classification, on the other hand, was performed by calculating the power density spectrum of the desired frequency. Specific thresholds were found for each user, and the final result was assigned based on this threshold. This classification approach resulted in a system that is both faster and more accurate than using one

single paradigm [17]. Both of these systems use one paradigm to strengthen the result of the other, first using EMG to boost MI performance, and second using ERP to boost the performance of SSVEP paradigm.

2.4 Paradigm Independent

Approaches described in previous sections require very specific procedures for integration of multiple paradigms. They are also limited by the use of only two paradigms each. This study proposes a new way of combining multiple BCI paradigms. The framework proposed in this study uses three paradigms without any paradigm-specific integration techniques such as those that were described in the previous sections (e.g. μ-gating, hierarchical priority-fusion). It works by running the EEG signal through each paradigm's preprocessing and CNN classification stage to extract the values at the final fully-connected layer for each of these CNNs. These vectors are concatenated, and the resulting feature vector is fed to the final classifier that decodes the user's intention. Such approach to paradigm combination is similar to paradigm extension, but it does not rely on paradigm-specific interactions such as μ-rhythm thresholding or pre-defined hierarchical priorities or other heuristics. All of it is learned. Development of this paradigm-independent framework is possible thanks to a big multi-subject multi-paradigm dataset, which is described in the next chapter [10].

Chapter 3

Data

3.1 Dataset description

A recently published sizeable BCI dataset [10] was used for this study. It boasts a large pool of subjects (54 participants), each of whom participated in multiple BCI paradigms. The EEG data for each participant was recorded with 62 Ag/AgCl electrodes. There are and three paradigms (ERP, MI, and SSVEP) recorded per user, with two sessions per paradigm. Same setup, without re-applying the EEG cap was used for recording all three paradigms. Due to some data inconsistencies (trigger setup), only 51 out of 54 subjects were used in this study. Experimental paradigms of ERP, MI, and SSVEP adhered to following BCI protocols: 6×6 row-column ERP speller, a binary MI task (right and left-hand imagery), and a four-class SSVEP system (5.45, 6.67, 8.57, and 12 Hz). Each session in the dataset is already divided in training and testing phases. Training samples were used to derive classifier models, while test samples were employed for performance evaluation. On a subject level, MI and SSVEP datasets consist of 100 training and 100 test samples, while the ERP dataset has 165 training and 180 test samples. Therefore, the total number of samples, concatenated across subjects, was 10200 (200×51), 17595 (345×51), and 10200 (200×51) for MI, ERP, and SSVEP, respectively.

Since each subject participates in three different paradigms, this dataset presents itself as very convenient in developing paradigm-independent approaches by allowing

classification between paradigms for individual subjects. It is also extremely useful for deep learning models, since the amount of data is quite extensive thanks to a large amount of subjects and trials.

3.2 Common data preprocessing

The same preprocessing steps were applied to the raw EEG data of all three paradigms. Firstly, EEG signals were down-sampled to 100 Hz and segmented from 0 to 4000 ms with respect to the stimulus onset. A baseline correction was applied by calculating the mean amplitudes from the -100 to 0 ms interval in each channel and subtracting them from the remaining time series. After preprocessing, EEG data had the following dimensions: (400 time points)×(62 channels)×(10200 trials·subjects) for MI and SSVEP who had the same dimensions and (400 time points)×(62 channels)×(17595 trials·subjects) for ERP trials. As a whole, 18615 samples were used for training, while 19380 samples were used for testing. Data preprocessing for individual paradigms is described in the following chapter.

Chapter 4

Paradigm-specific feature representations

Paradigm-specific pre-processing steps are outlined in this chapter. Each paradigm elicits its own neural response and thus requires some specific steps to boost its signal-to-noise ratio. For example MI and SSVEP features mostly rely on a frequency domain, while ERP features are most discriminant in the temporal domain. The pre-processing steps outlined below are introduced to accurately classify the three individual paradigms as well as inner classes of each paradigm. These approaches were selected since they result in 2-dimensional representation of data that is suitable to be fed into a CNN model, while retaining or boosting the much needed discriminant features.

To clarify the notation used in this chapter and further, let us denote $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n} \in \mathbb{R}^{T \times C}$ as a set of EEG samples for a single paradigm, where $\mathbf{x}_i$, T, C, and n indicate a feature map for a single trial, time points, number of channels, and number of EEG samples per subject, respectively. $\mathbb{Y}$ is used to denote the set of corresponding class labels. $\mathbf{X}_e$, $\mathbf{X}_m$, $\mathbf{X}_s$ are defined to denote the initial datasets for three individual paradigms, i.e., *ERP*, *MI*, and *SSVEP*, respectively. Next, to discriminate between-paradigm and within-paradigm labels, $\mathbb{Y}_b$ and $\mathbb{Y}_w$ are introduced. Specifically, $\mathbb{Y}_b \in \{ERP, MI, SSVEP\}$ denotes the class labels for three paradigms while $\mathbb{Y}_w \in \{ERP, MI\text{-}right, MI\text{-}left, 5.45\ Hz, 6.67\ Hz, 8.57\ Hz, 12\ Hz\}$ denotes the

within-paradigm classes.

4.1 ERP-relevant temporal feature generation

Minimal pre-processing was performed to generate ERP-relevant features. The dataset $\mathbf{X}$ was bandpass filtered between 0.1 and 40 Hz to get rid of the high-frequency noise present in the signal. Since the ERPs could be detected throughout the entire duration of a trial and throughout both visual and frontal cortexes, no channel or time-interval selection was performed. To prepare the dataset before passing it through the CNN, each channel was normalized to zero mean and unit variance. This resulted in a matrix of 400 time points by 62 channels for each trial. The resulting dataset is denoted as $\mathbb{X}_e$.

4.2 MI-relevant spatial-spectral feature generation

Previous studies presented a mix of approaches to extract MI-relevant features. Since ERDs occur in different frequency bands for different users, most of them are based on finding certain user-specific or user-invariant spatial-spectral filters [8, 18, 9]. A heuristic approach that employs pre-set single or multiple band-pass filters, has been widely used in BCI competitions [8]. However, this approach does not generalize well and has limited performance for illiterate users [10]. Mutual-information based optimization approaches have shown great performance but require a lot of computational complexity which may lead to over-fitting [19]. This study mixes both approaches, and uses pre-selected user-invariant filters along with pre-selected mutual-information based filters that fit most subjects well.

In a previous study on the same dataset, the mutual information I was calculated for select frequency bands where each frequencies had their own contribution scores to the final classification. The constructed filter-bank $\mathbf{b}$ is defined by combination of well-known standard bands (θ, μ, and β) with addition of the top-performing frequency bands from the previously mentioned study [18], $\mathbf{b} = \{[\theta], [\mu], [\beta], [rank1], [rank2]..\}$.

This resulted in a following filter-pool: $([4,\ 8],[8,\ 13],[13,\ 30],[7,\ 9],[5,\ 10],[26,\ 34],[11,\ 13])$. Band-pass filtered data for each frequency band $\mathbf{b}_f$ was then obtained and denoted as $\mathbb{Z}_f$, i.e., $\mathbb{Z}_f$=$\mathbf{b}_f \bigotimes \mathbf{X}_M$.

Common Spatial Patterns (CSP) is a spatial filtering and dimensionality reduction technique that maximizes the variance of one class, while simultaneously minimizing the variance of the other class [20]. CSP transformation for class c can be calculated by the following equation:

$$\mathbf{W_c} = \arg\max_{\mathbf{W}} \frac{\mathbf{W}^{\mathrm{T}} \Sigma_c \mathbf{W}}{\mathbf{W}^{\mathrm{T}} (\Sigma_1 + \Sigma_2) \mathbf{W}} \tag{4.1}$$

where Σ_1 and Σ_2 denote the covariance matrices for each class.

CSP was applied to the band-pass filtered data $\mathbb{Z}$ to find the projection matrix $\mathbf{W} \in \mathbb{R}^{U \times U}$, which is obtained by solving a generalized eigenvalue decomposition, where U is the number of spatial filters to be pairwise selected from the CSP.

The spectrally and spatially filtered EEG data is acquired based on the filter-bank. Since spatial filters $\mathbf{W}$ are unique for each subject, corresponding trials were multiplied by corresponding subjects' spatial filter. The sets of spatially and spectrally optimized EEG data were denoted as $\mathbf{X}_M^f = \mathbf{W}_f \cdot \mathbb{Z}_f$, where a single frequency dataset (e.g., $\mathbf{X}_M^1$) is formed as $T \times U \times n$ corresponding to the time points, number of spatial filters, and number of samples. Since we consider a filter-bank approach, $\mathbb{X}_e$ was constructed by stacking of individual $\mathbf{X}_M^f$ along the spatial dimension, thereby forming the $\mathbb{X}_e$. The dataset is then normalized before feeding it to the CNN. The resulting trials have following dimensions: 400 time points by 28 channels, where 28 is formed by stacking spatially-filtered data seven times, one for each filter-band.

4.3 SSVEP-relevant spectral feature generation

Previous SSVEP classification approaches have generated a time-frequency matrix using Fourier transform (FFT) or continuous wavelet transform (CWT) and fed it into the CNN network [21]. These methods showed great performance, but required high computational complexity.

This study employed multi-channel CCA [11, 22] to acquire frequency information and to generate powerful spectral features. Canonical correlation analysis (CCA) is a statistical approach that allows us to measure the correlation between the EEG signal $\mathbf{x}$ and artificially generated reference signals $\mathbf{s}$. From linear combinations $\mathbf{X} = \mathbf{x}^T\mathbf{w}_x$ and $\mathbf{S} = \mathbf{s}^T\mathbf{w}_s$, CCA finds the weight vectors, $\mathbf{w}_x$ and $\mathbf{w}_s$, which maximize the correlation between $\mathbf{X}$ and $\mathbf{S}$ by solving the following equation:

$$\rho(\mathbf{X}, \mathbf{S}) = \underset{\mathbf{w_x}, \mathbf{w_s}}{\arg\max} \frac{E[\mathbf{X} \cdot \mathbf{S}^T]}{\sqrt{E[\mathbf{X} \cdot \mathbf{X}^T]E[\mathbf{S} \cdot \mathbf{S}^T]}} \tag{4.2}$$

The reference signal $\mathbf{s}_{fn}$ is defined as:

$$\mathbf{s}_{fn} = \begin{bmatrix} \sin(2\pi \cdot f_n \cdot t) \\ \cos(2\pi \cdot f_n \cdot t)) \\ \vdots \\ \sin(2\pi \cdot (f_n \cdot N_h) \cdot t)) \\ \cos(2\pi \cdot (f_n \cdot N_h) \cdot t)) \end{bmatrix}, t = \left[\frac{1}{f_n}, \frac{2}{f_n}, \ldots, \frac{T}{f_n}\right] \tag{4.3}$$

where f_i represent our SSVEP target frequencies (f_1=12 Hz, f_2=8.57 Hz, f_3=6.67 Hz, and f_4=5.45 Hz), and T and N denote data points and number of harmonics respectively. Ten channels in the occipital cortex were selected (P-7/3/z/4/8, PO-9/10, and O-1/z/2) and ρ canonical correlations were calculated and stacked across individual target frequencies and selected channels. The resulting dataset $\mathbf{P}_w$ is formed as $(f_n \times N_h) \times ch$ corresponding to number of frequencies, harmonics, and channels, respectively. ρ values of SSVEP data have a distinctive distribution, where one single value $\rho_{\hat{f}}$ is significantly higher than the others ρ_f. The reason behind this is that only a target frequency is greatly elevated by the CCA algorithm. On the contrary, this type of pattern cannot be observed for MI or ERP derived CCA signals since these specific frequency components are not a significant part of their signal.

While $\mathbf{P}_w$ is primarily used for decoding the within-paradigm classes of the SSVEP paradigm, a second matrix $\mathbf{P}_b$ is defined to introduce additional features that help in distinguishing the SSVEP paradigm from the other two paradigms. For between-

paradigm decoding, ρ canonical correlation is calculated across all channels and target frequencies. This results in a 4×62 matrix. $\mathbf{P}_b$ is obtained by calculating the standard deviation of this matrix along each frequency, resulting in a 1×62 matrix, which then is reshaped to 8×8 with two additional blank entries, so that it could be concatenated with $\mathbf{P}_w$. This results in the matrices shown here:

$$\mathbf{P}_w = \begin{pmatrix} \rho_{(ch_1,f_1)} & \cdots & \rho_{(ch_1,f_{n\times h})} \\ \rho_{(ch_2,f_1)} & \cdots & \rho_{(ch_2,f_{n\times h})} \\ \vdots & \ddots & \vdots \\ \rho_{(ch_{10},f_1)} & \cdots & \rho_{(ch_{10},f_{n\times h})} \end{pmatrix}, \mathbf{P}_b = \begin{pmatrix} \sigma_{\rho(ch1,f_n))} & \cdots & \sigma_{\rho(ch4,f_n))} \\ \sigma_{\rho(ch5,f_n))} & \cdots & \sigma_{\rho(ch8,f_n))} \\ \vdots & \ddots & \vdots \\ \sigma_{\rho(ch61,f_n))} & \cdots & null \end{pmatrix} \tag{4.4}$$

Finally, the $\mathbb{X}_s$ dataset is constructed by concatenating $\mathbf{P}_w$ and $\mathbf{P}_b$ (see Figures 4.3.1 and 4.3.2). The dataset was normalized before using it for CNN training. The resulting size of the CCA feature is 8×10 for each trial.

Three domain-specific feature maps ($\mathbb{X}_e$, $\mathbb{X}_m$, and $\mathbb{X}_s$) are used in the CNN classification. To optimize the convergence of the deep learning approach, all three of these datasets were normalized to have zero mean and unit variance. These three feature maps are fed to the CNNs in a specific way that is outlined in the next chapter.

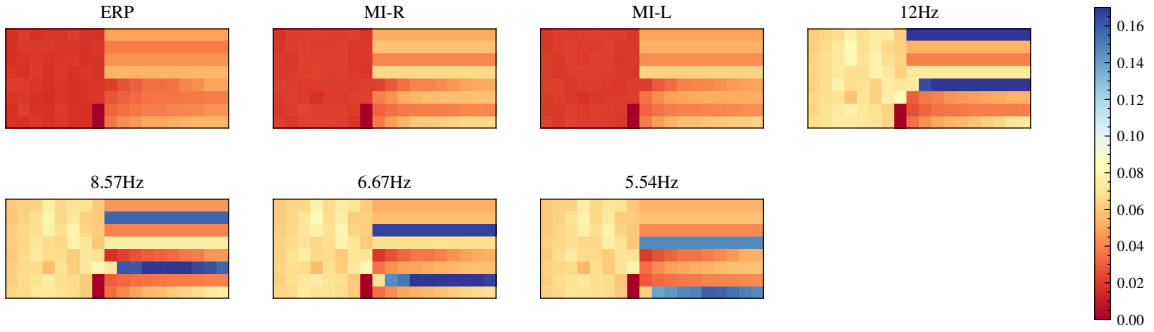


Figure 4.3.1: SSVEP-relevant spectral feature representation averaged from the training set. CCA features to distinguish SSVEP and [ERP/MI] ($\mathbf{P}_b$) as well as to decode individual within-SSVEP classes ($\mathbf{P}_w$) were combined as 8×18 matrix. There are clear high-magnitude lines corresponding to each target frequency and their harmonic.

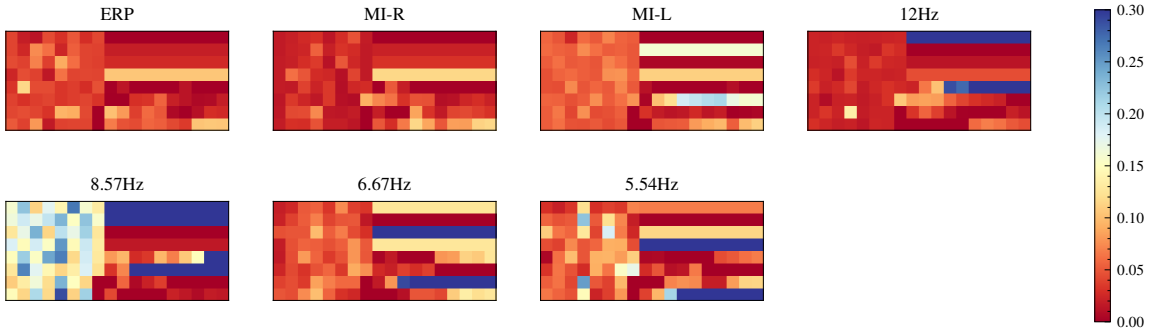


Figure 4.3.2: Random SSVEP-relevant feature sample from the training set. As it can be seen from this sample, the high-magnitude lines are not always clear.

Chapter 5

Classification

5.1 CNN structure

This study proposes an extendable CNN framework that allows classification between paradigms and their constituent classes. Hierarchical classification is a common strategy, especially so when the classes are categorized into subgroups, and have been shown to lead to greater performance when compared to standard CNNs [23]. However, hierarchical solutions rely on pre-determined order of comparisons, that is usually determined heuristically by experts. This heuristic can be omitted if another classifier is used in its place. Three convolutional neural networks are constructed independently: *eCNN*, *mCNN*, and *sCNN* which are specified for learning domain-specific information. Each of these CNN models is responsible for its own inner-paradigm classification. These CNN models are used to extract paradigm-specific feature vectors. These vectors are then concatenated to get the full representation of the input trial. The concatenated vector is then used to carry out the final classification. The CNN sub-models and final meta-classifier are trained separately during the training stage, but constitute a single model during inference. The framework assumes that only one paradigm can be used at a time, resulting in an output of one of seven classes: ERP, MI-left, MI-right, 12Hz, 8.57Hz, 6.67Hz, 5.54Hz. It is shown on Figure 5.1.1.

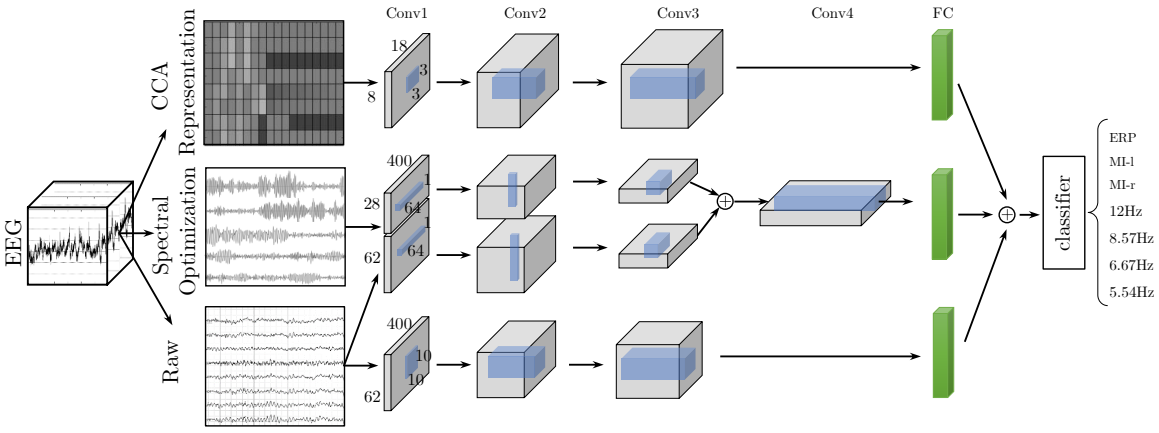


Figure 5.1.1: Paradigm-independent framework for discerning between ERP, MI, and SSVEP paradigms as well as between the within-paradigm classes. Three individual CNN networks (sCNN, mCNN, and eCNN) were used to extract the final vector used in the classification.

5.1.1 ERP CNN

The eCNN model is constructed from three convolutional layers, each with zero-padding to preserve the dimensions. Each of the convolutional blocks includes padding, convolution, batch-normalization and ReLU is chosen as an activation function. All blocks except for the final one include max pooling, to reduce the dimensionality. Finally, a fully-connected layer is added, and eCNN sub-model is trained to distinguish: *ERP* vs. *non-ERP*. Only one inner-class was chosen to represent the ERP paradigm, since having both target and non-target classes would interfere with other paradigm decoding, because non-target ERP would overlap with other paradigms' inner classes. The architecture of this model is shown in detail on Table 5.1.1. This CNN setup with 3 convolutional layers and a filter size of 10x10 reached a validation accuracy of 98%. Similar architecture with 5x5 filter size reached a lower accuracy of 96%. A CNN network with two convolutional layers resulted in a validation accuracy of 96%. Using four convolutional layers resulted in 98% accuracy, however it also required a significantly longer training time, so it was not used.

Layer	*Type*	*#Filters*	*Kernel*	*Stride*	*Output*
1	Input				1x400x62
	Conv2D	8	10x10		8x400x62
	BatchNorm				
	ReLU				
	MaxPool2D		2x2	2x2	8x200x31
2	Conv2D	16	10x10		16x200x31
	BatchNorm				
	ReLU				
	MaxPool2D		2x2	2x2	16x100x15
3	Conv2D	32	10x10		32x100x15
	BatchNorm				
	ReLU				
	Linear				1x2

Table 5.1.1: eCNN model architecture.

5.1.2 MI CNN

Both raw($\mathbb{X}_e$) and spectrally-spatially optimized signal($\mathbb{X}_m$) were used in training the mCNN model. To efficiently extract both temporal and spectral features from $\mathbb{X}_e$ and $\mathbb{X}_m$, the convolutional model from [24] was applied to the signals. Each convolutional layer had its role. First layer uses a temporal convolution to extract the frequency content of the signal. Second layer uses a depthwise convolution to learn separate spatial filters for each resulting frequency. The separable convolution is then used to extract the summary temporal data from each channel and then mix them together [24]. Outputs of the third convolutional layer were concatenated channel-wise. This allows the network to extract relevant features from both $\mathbb{X}_e$ and $\mathbb{X}_m$. 10% dropout layers were added after the convolutions to avoid overfitting. Finally, a fully-connected layer is added. The model uses exponential linear unit (ELU) as its activation function, since it was shown that it results in a better performance [24]. This part of the model is trained to distinguish between *MI-left*, *MI-right* and *non-MI*. The network architecture is shown on table 5.1.2. At first the architecture proposed in [24] was used with a temporal convolution block at the end instead of the regular one, this resulted

in a validation accuracy of 93%, using a simple convolutional block in the fourth layer boosted the validation accuracy to 95%. Single-branch CNN model, showed a validation accuracy of 86%. Finally it was decided to use a double-branch model, with a convolutional layer at the end.

Layer	***Type***	***#Filters***	***Kernel***	***Value***	***Output***
1	Input				1x62x400, 1x28x400
	Conv2D	4	1x64		4x62x400, 4x28x400
	BatchNorm				
2	DepthwiseConv2D	4*2	62x1, 28x1		8x1x400
	BatchNorm				
	ELU				
	AvgPool2D		1x8		8x1x50
	Dropout			10%	
3	SeparableConv2D	8	1x16		8x1x50
	BatchNorm				
	ELU				
	AvgPool2D		1x8		8x1x6
4	Concatenate				16x1x6
	Conv2D	32	1x3		32x1x6
	BatchNorm				
	ELU				
	Dropout			10%	
	Linear				1x3

Table 5.1.2: mCNN model architecture. Since first 3 layers were used for two separate data streams, their respective values are separated by commas. Values that are not separated by commas are the same for both streams.

5.1.3 SSVEP CNN

Since the input dataset $\mathbb{X}_s$ is relatively low-dimensional, the sCNN network is comparatively simple. Because the main objective in this classification is to find lines of high activation in the picture, as shown in Figure 4.3.2, a small-scale kernels were used in the network. Finally, a fully-connected linear layer is added. This sub-model is trained

Layer	*Type*	*#Filters*	*Kernel*	*Value/Stride*	*Output*
1	Input				1x18x8
	Conv2D	8	3x3		8x18x8
	BatchNorm				
	ReLU				
	Dropout			20%	
	MaxPool2D		3x3	2	8x8x3
2	Conv2D	8	4x4		8x8x3
3	BatchNorm				
	ReLU				
	MaxPool2D		3x3	2	8x3x1
	Conv2D	16	8x8		16x3x1
	BatchNorm				
	ReLU				
	Linear				1x5

Table 5.1.3: sCNN model architecture. Zero-padding was used to maintain the feature map dimensions.

to distinguish between *non-SSVEP*, and *12 Hz*, *8.57 Hz*, *6.67 Hz*, and *5.45 Hz*. Two and three layer CNN variants were tested, both resulting in the validation accuracy of 96%. Since the input map is already low-dimensional, adding additional layers would be useless. Three-layer variant was used for the final training because of its faster training time (lower amount of parameters in the fully-connected layer). The sCNN network architecture is shown in Figure 5.1.3.

PyTorch framework was used for training the models. The training dataset was divided into training and validation sets in 75:25 ratio, preserving the class proportions. This resulted in 13961 samples in the training set and 4654 samples in the validation set. Validation set was used to test differing architectures. After the architecture was decided on, the model was trained on all of the initial training dataset. All three networks were trained by minimizing the cross-entropy loss using Adam optimizer with a learning rate of 0.001 for 100 epochs. The training is conducted separately for each model. PyTorch's weighted sampler was used to avoid class-imbalance problems. After the training of each of the individual sub-model is completed, their fully-connected layer activations are collected and concatenated for the entire train dataset (with

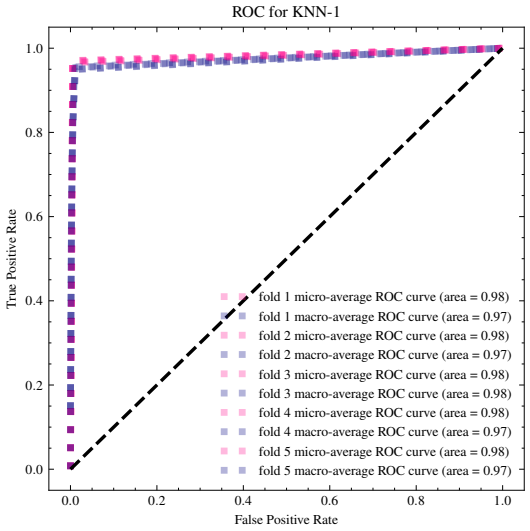
exceptions covered in the next chapter). This resulting dataset of concatenated feature vectors is then used to train a meta-classifier (pictured on the far-right of Figure 5.1.1).

The framework's design relies on the ability of each paradigm's CNN sub-model to classify the sub-classes of this paradigm and an additional class corresponding to signals that don't belong to this paradigm. Essentially, the sub-models should be able to determine if the incoming signal is within their respective paradigm or not and determine which in-paradigm class it belongs to.

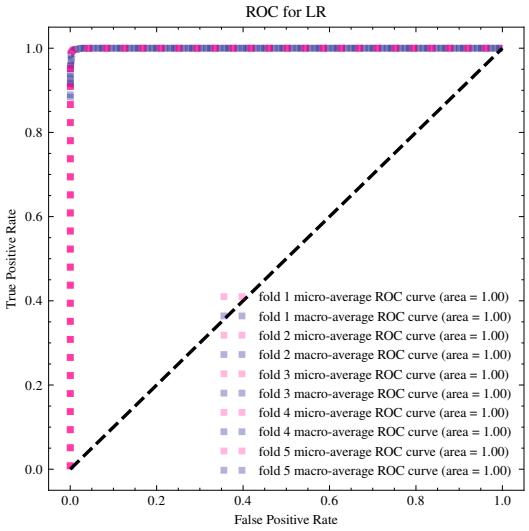
5.2 Meta-classifier

In the final classification step, softmax layers of each CNN are discarded. Here we are only interested in the raw feature vectors that come out of the last fully connected layer of each CNN. The CNNs trained in the last section are used as feature extractors to convert the training dataset into a new training "meta-dataset" by concatenating the raw feature vector outputs of each CNN. Resulting dataset is used to train a meta-classifier. The term meta-classifier is used since this approach is similar to stacking classifier, which uses the outputs of other classifiers, called "meta-features", to train a final estimator.

A number of classification algorithms were compared with the intention to select the best one as the meta-classifier. Common estimators such as LDA, KNN, SVM, Logistic Regression, Naive Bayes, Decision Tree, Random Forest and Neural Network were compared by running 10-fold cross-validation on the training "meta-features". Since most of the classification already occurs in the sub-models all meta-classifier candidates performed exceptionally well. The classifier with the weakest performance still achieves ROC-AUC of around 98%, while most other classifiers achieved ROC-AUC of 100% as seen in Figure 5.2.1. Table 5.2.1 and Figure 5.2.2 present the results of this comparison. Logistic Regression was found to be the most accurate estimator at 98.7$\pm$0.2% accuracy and was chosen as the meta-classifier.



(a) ROC for KNN with one nearest neighbor.



(b) ROC for Logistic Regression

Figure 5.2.1: ROC curves. Since the classifiers' prediction results in non-binary amount of classes, micro and macro averaging were used to calculate ROC.

Estimator	**LR**	**LDA**	**KNN-5**	**KNN-3**	**KNN-1**	**CART**
Accuracy	0.987430	0.972388	0.973194	0.973194	0.968520	0.978673
std	0.002403	0.003775	0.002731	0.002731	0.002521	0.003078
Estimator	**RF**	**NB**	**SVM**	**FC-20x2**	**FC-20**	**FC-40**
Accuracy	0.986355	0.975558	0.985711	0.986248	0.986893	0.986516
std	0.002025	0.004331	0.002255	0.002243	0.002622	0.001788

Table 5.2.1: Comparison of classification algorithms. With standard deviations.

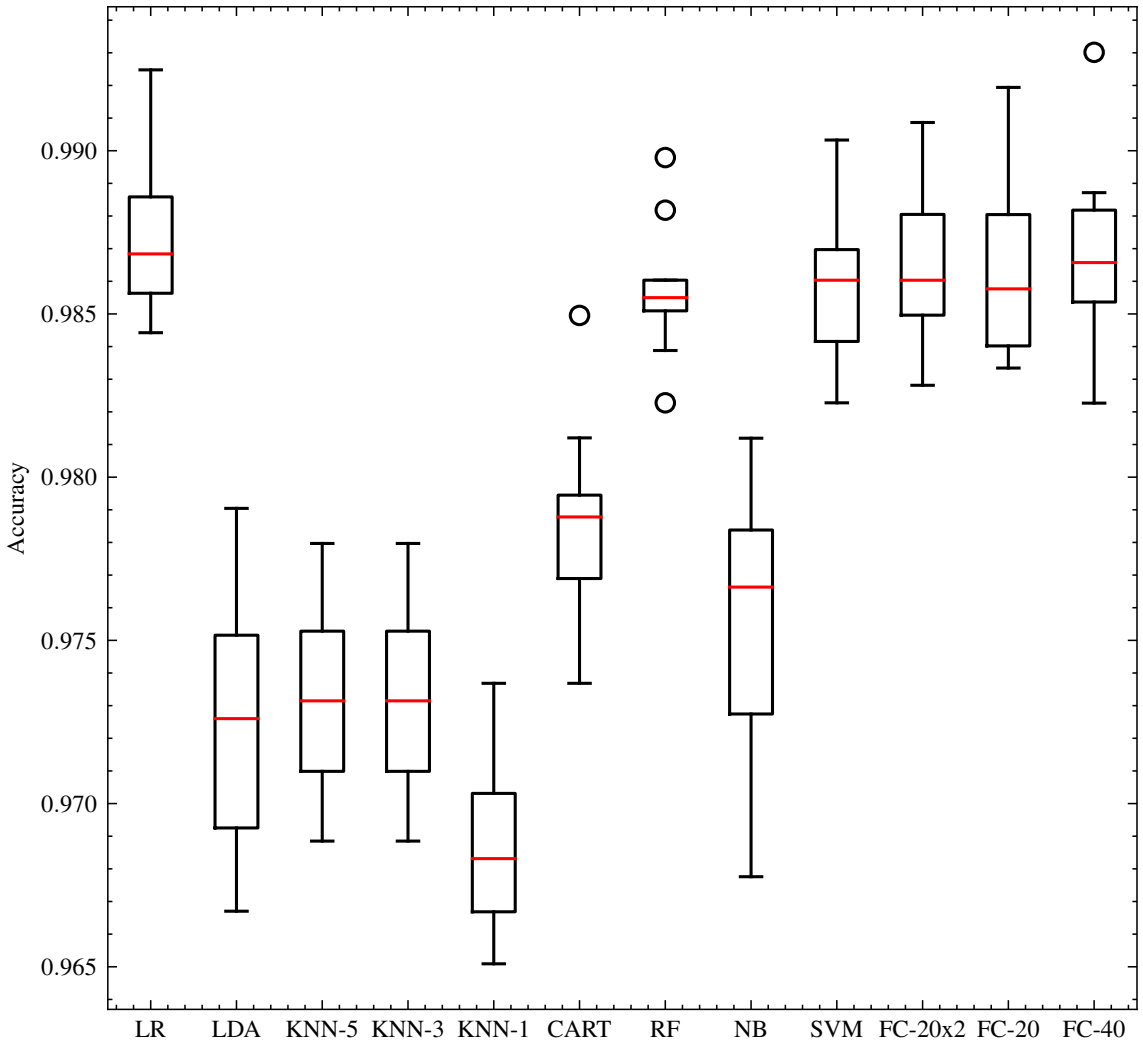


Figure 5.2.2: A box plot comparison of classification algorithms obtained from 10-fold cross-validation over the train data. From left to right: logistic regression, LDA, KNN with 5,3,1 nearest neighbors, decision trees, random forest, naive bayes, svm, two fully connected layers with 20 neurons each, single 20 neuron layer and a single 40 neuron layer.

Chapter 6

Results

6.1 Performance evaluation

Due to differences in the brain signal characteristics of different users, some BCIs struggle with subject-dependency problem. This problem occurs when the BCI is tested on a data from a previously unseen subject, leading to low accuracy. To check for this problem, the performance of this framework was evaluated by considering both subject-dependent and subject-independent cases (pictured in Figure 6.1.1). For the subject-independent approach, all of the data coming from a subject was dropped before training the classification models corresponding to this subject, so that the model would not be familiar with the neural patterns produced by this subject. This leave-one-out approach required training of 153 sub-models (51 of each eCNN, mCNN, sCNN) and 51 meta-classifiers. The models were then assessed subject-wise on the following metrics: 7-class within-paradigm classification (deciding both the paradigm and the inner class), 3-class between-paradigm classification (deciding paradigm only).

6.2 Results

For between-paradigm decoding, the proposed framework showed an accuracy of 97.51% in both subject-independent and subject-dependent cases. It showed 97.48% accuracy in decoding ERP paradigm in subject-dependent case and 98.46% in subject-

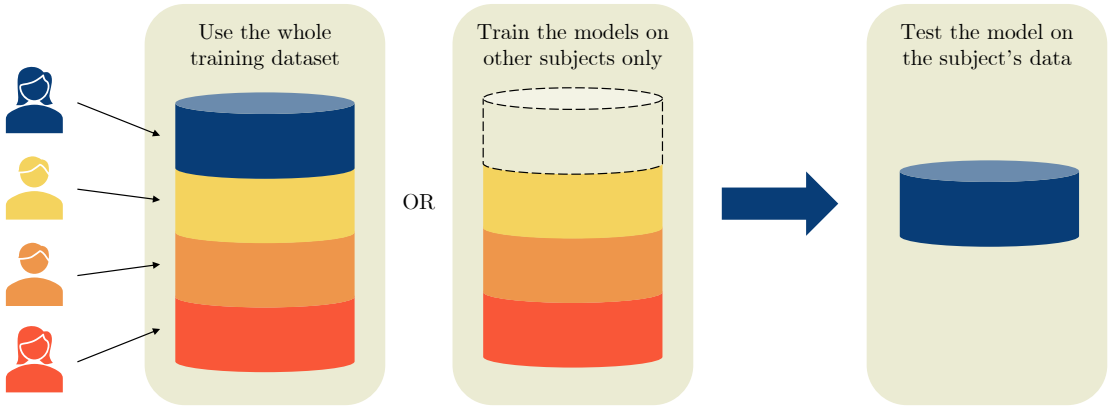


Figure 6.1.1: Two variants of training. Subject-dependent training, where all of the dataset is used, and subject-independent training, where only the data of other subjects is used for training. Testing is similar in both cases.

		Linear Classifier [12]	Linear Classifier (Hierarchical)[12]	**Proposed Method (All data)**	**Proposed Method (Subject-independent)**
Paradigm Decoding	ERP	64%	84.1%	97.48±4.21%	98.46±1.96%
	MI	59%	78.5%	95.57±11.67%	93.59±12.69%
	SSVEP	58%	89%	99.51±1.09%	99.73±0.91%
	Total	73.44±8.73	73.44±8.73	97.51±3.61%	97.51±3.39

Table 6.2.1: Comparison of paradigm decoding between the previous linear classifier approach and deep-learning based framework proposed in this study.

independent case. For MI, It showed 95.57% accuracy in subject-dependent case and 93.59% accuracy in subject-independent case, in both cases the standard deviation is quite high at over 11.67% for subject-dependent and 12.69% subject independent case. The SSVEP decoding for subject dependent approach is very accurate at 99.51%, its even more accurate for subject-independent case, with accuracy of 99.73%. The values can be seen in Table 6.2.1, with comparison to a previous RLDA based linear classifier approach. Two additional subjects were used in comparison with the study presented in the Table 6.2.1, subjects 48 and 50. Calculating the total accuracy of the framework without those subjects results in similar $97.51 \pm 3.43\%$ accuracy in subject-independent and $97.51 \pm 3.65\%$ in subject-dependent cases. Additional subjects slightly helped in reducing the standard deviation. Tables A.0.2 and A.0.4 in the appendix provide the accuracies for each subject in the dataset.

Within-paradigm decoding, on the other hand reached an accuracy of 87.9±4.51% in subject-independent case and 87.6±5.59% in subject-dependent case. Paradigm-

wise, ERP decoding is the same as in between-paradigm approach, since ERP only has one class. MI decoding showed a low average accuracy of 60.47±14.27% in the subject-independent case and 62±14.72% in the subject-dependent case. The model classified SSVEP with 96.31±7.28% accuracy in the subject-independent case, and 95.41±8.18% accuracy in the subject-dependent case. The full distribution of within-paradigm accuracies per subject can be observed in Tables A.0.1 and A.0.3.

A comparison between the accuracies of within-paradigm classification and between-paradigm classification is plotted on Figure 6.2.2. It is notable that for all subjects, except subject 23 and subject 51, the within-paradigm classification scores are above 80%. Figure 6.2.1 presents a confusion matrix for subject-independent paradigm-independent classification. It shows contributions of all label classification accuracies to the final accuracy. It is notable that most of the accuracy (8.72%) was lost in the within-paradigm MI classification.

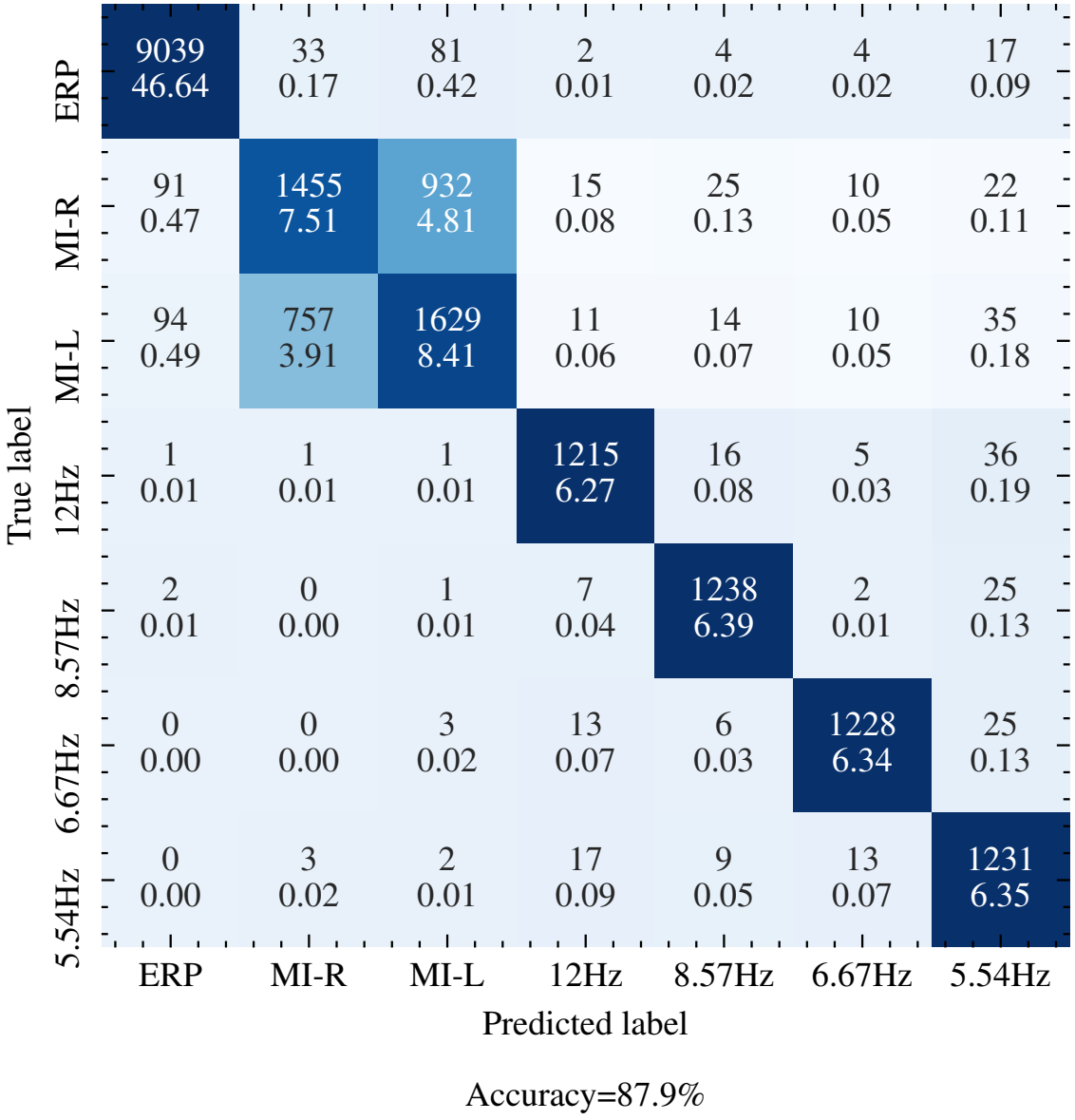


Figure 6.2.1: Confusion matrix of 7-class classification for the subject-independent model. Rows and columns in the matrix indicate predicted class (classifier output) and true class (label).

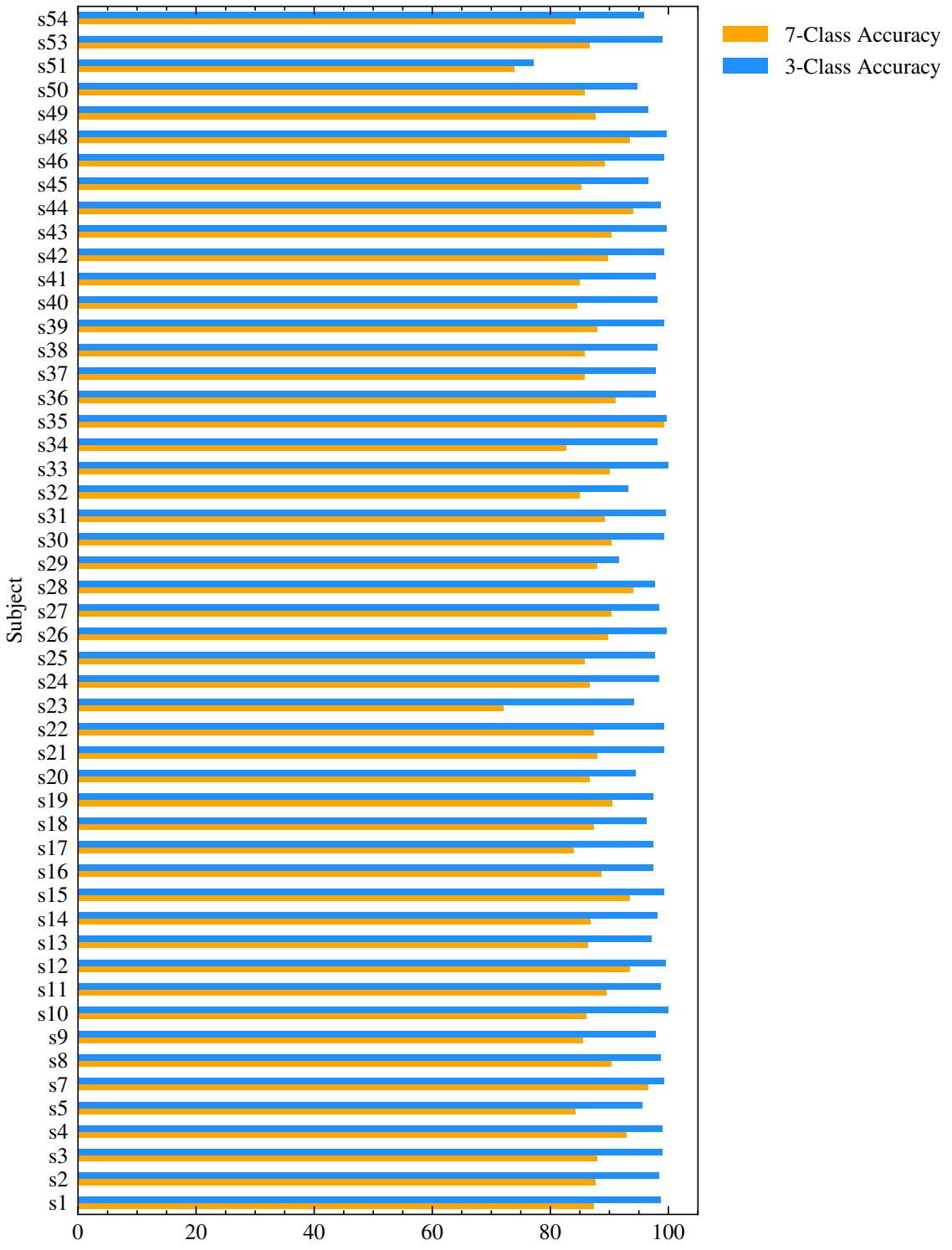


Figure 6.2.2: Performance comparison between 7-class and 3-class subject-independent classification accuracies per subject.

Chapter 7

Discussion and Conclusion

This study proposed a novel framework for constructing paradigm-independent brain-computer interfaces. Each single paradigm's shortcomings could be alleviated by introducing another paradigms, which allows us to increase the available amount of classes and to potentially sidestep the negative aspects of individual paradigms such as fatigue, illiteracy, etc. This presents a great potential in developing paradigm-independent BCIs. The framework introduced in this study allows researchers to use already implemented CNN models for individual paradigms and combine them with minimal fine-tuning, required to incorporate an additional "non-paradigm" class to each network's output. The framework is modular and allows users to swap the individual sub-models with updated state-of-the-art solutions, only a retraining of the meta-classifier is needed in the end.

The proposed framework showed great accuracy in decoding both between individual paradigms and within paradigm classes. One weakness of the proposed solution is low-performance MI sub-model, that showed only about 60% accuracy on average. It could potentially be swapped with more accurate state-of-the-art MI classification models such as [18, 25, 26], or subject-specific optimizations could also be introduced. The 7-class paradigm-independent model allows users to perform a higher amount of actions and encode a higher amount of intentions than usual unimodal BCI paradigms. It opens the road to designing more complex BCI interfaces. One potential use for this 7-class BCI would allow users to play NES games, where each of 7 classes could be

mapped to the keys of NES controller, e.g. SSVEP classes to represent D-pad actions, MI classes for A and B buttons and ERP could be used for Start and Select.

The framework could be used to design novel highly accurate brain computer interfaces based on between-paradigm classification, since even for the subjects whose within-paradigm MI classification accuracy was around 50% such as subjects 45 and 34, between-paradigm classification is significantly higher at 89% and 99% respectively (in subject-independent classification). This framework would potentially allow MI-illiterate users to still utilize MI paradigm, even if it would only introduce a single additional class. The subject-independent 3-Class classification showed exceptional 97.5% accuracy, which would mean that paradigm-independent approach is a perfect fit for subject-independent BCIs, where the model could only be trained once, without fine-tuning to each individual new subject.

A potential drawback of this solution is that it was not tested in the real world. The dataset it was trained on was recorded in separate bursts, where paradigms were used one at a time. Because of it, in the real world, there might be some issues in the classifiers, where paradigms could interfere with each other. For example, in the game of tetris described in [15], if additional inputs were introduced through use of an ERP paradigm, occasional ERPs might get triggered by users waiting for a certain tetromino piece. In the NES controller example, the ERPs could also be elicited by a possible unexpected occurrence in the videogame. Examples like these are the reason why this framework needs to be tested in the real world, and might need some problem-specific tweaks to avoid paradigm misfiring.

Broader Impact

This study proposed a paradigm-independent model construction framework that could easily be extended and improved with better models and additional paradigms. The trained model showed promising accuracy on both between and within-paradigm classification problems. The paradigm-independent approach has high potential of being a preferred solution for subject-independent BCIs.

Appendix A

Tables

Table A.0.1: Within-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
s1	97.777778	62.000000	94.000000	87.368421
s2	100.000000	53.000000	100.000000	87.631579
s3	100.000000	66.000000	88.000000	87.894737
s4	98.888889	75.000000	100.000000	92.894737
s5	91.111111	57.000000	99.000000	84.210526
s7	100.000000	87.000000	100.000000	96.578947
s8	99.444444	76.000000	88.000000	90.263158
s9	100.000000	47.000000	98.000000	85.526316
s10	100.000000	53.000000	94.000000	86.052632
s11	98.333333	63.000000	100.000000	89.473684
s12	100.000000	75.000000	100.000000	93.421053
s13	100.000000	49.000000	99.000000	86.315789
s14	98.333333	53.000000	100.000000	86.842105
s15	100.000000	80.000000	95.000000	93.421053

Continued on next page

Table A.0.1: Within-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
s16	98.333333	61.000000	99.000000	88.684211
s17	98.888889	46.000000	95.000000	83.947368
s18	94.444444	63.000000	99.000000	87.368421
s19	99.444444	66.000000	99.000000	90.526316
s20	100.000000	50.000000	99.000000	86.578947
s21	100.000000	54.000000	100.000000	87.894737
s22	100.000000	53.000000	99.000000	87.368421
s23	97.777778	47.000000	51.000000	72.105263
s24	97.222222	59.000000	95.000000	86.578947
s25	99.444444	48.000000	99.000000	85.789474
s26	100.000000	65.000000	96.000000	89.736842
s27	97.777778	67.000000	100.000000	90.263158
s28	95.555556	89.000000	96.000000	93.947368
s29	92.222222	70.000000	98.000000	87.894737
s30	100.000000	63.000000	100.000000	90.263158
s31	99.444444	60.000000	100.000000	89.210526
s32	97.222222	53.000000	95.000000	85.000000
s33	100.000000	65.000000	97.000000	90.000000
s34	96.666667	51.000000	89.000000	82.631579
s35	99.444444	99.000000	99.000000	99.210526
s36	96.111111	73.000000	100.000000	91.052632
s37	97.222222	53.000000	98.000000	85.789474
s38	97.777778	50.000000	100.000000	85.789474
s39	99.444444	58.000000	97.000000	87.894737

Continued on next page

Table A.0.1: Within-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
s40	98.333333	54.000000	90.000000	84.473684
s41	99.444444	45.000000	99.000000	85.000000
s42	98.888889	68.000000	95.000000	89.736842
s43	99.444444	64.000000	100.000000	90.263158
s44	97.777778	81.000000	100.000000	93.947368
s45	98.888889	50.000000	96.000000	85.263158
s46	99.444444	62.000000	98.000000	89.210526
s48	99.444444	76.000000	100.000000	93.421053
s49	98.888889	57.000000	98.000000	87.631579
s50	99.444444	47.000000	100.000000	85.789474
s51	100.000000	6.000000	95.000000	73.947368
s53	98.888889	52.000000	99.000000	86.578947
s54	94.444444	63.000000	87.000000	84.210526
mean	98.464052	60.470588	96.313725	87.899897
stdev	1.963687	14.273729	7.282776	4.508452

Table A.0.2: Between-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
s1	97.777778	99.000000	100.000000	98.684211
s2	100.000000	94.000000	100.000000	98.421053
s3	100.000000	96.000000	100.000000	98.947368

Continued on next page

Table A.0.2: Between-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
s4	98.888889	98.000000	100.000000	98.947368
s5	91.111111	99.000000	100.000000	95.526316
s7	100.000000	97.000000	100.000000	99.210526
s8	99.444444	96.000000	100.000000	98.684211
s9	100.000000	92.000000	100.000000	97.894737
s10	100.000000	100.000000	100.000000	100.000000
s11	98.333333	98.000000	100.000000	98.684211
s12	100.000000	98.000000	100.000000	99.473684
s13	100.000000	89.000000	100.000000	97.105263
s14	98.333333	96.000000	100.000000	98.157895
s15	100.000000	98.000000	99.000000	99.210526
s16	98.333333	94.000000	99.000000	97.368421
s17	98.888889	92.000000	100.000000	97.368421
s18	94.444444	96.000000	100.000000	96.315789
s19	99.444444	91.000000	100.000000	97.368421
s20	100.000000	79.000000	100.000000	94.473684
s21	100.000000	97.000000	100.000000	99.210526
s22	100.000000	98.000000	99.000000	99.210526
s23	97.777778	82.000000	100.000000	94.210526
s24	97.222222	99.000000	100.000000	98.421053
s25	99.444444	92.000000	100.000000	97.631579
s26	100.000000	100.000000	99.000000	99.736842
s27	97.777778	98.000000	100.000000	98.421053
s28	95.555556	100.000000	99.000000	97.631579

Continued on next page

Table A.0.2: Between-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
s29	92.222222	83.000000	99.000000	91.578947
s30	100.000000	97.000000	100.000000	99.210526
s31	99.444444	99.000000	100.000000	99.473684
s32	97.222222	79.000000	100.000000	93.157895
s33	100.000000	100.000000	100.000000	100.000000
s34	96.666667	99.000000	100.000000	98.157895
s35	99.444444	100.000000	100.000000	99.736842
s36	96.111111	99.000000	100.000000	97.894737
s37	97.222222	97.000000	100.000000	97.894737
s38	97.777778	97.000000	100.000000	98.157895
s39	99.444444	98.000000	100.000000	99.210526
s40	98.333333	96.000000	100.000000	98.157895
s41	99.444444	93.000000	100.000000	97.894737
s42	98.888889	99.000000	100.000000	99.210526
s43	99.444444	100.000000	100.000000	99.736842
s44	97.777778	99.000000	100.000000	98.684211
s45	98.888889	89.000000	100.000000	96.578947
s46	99.444444	98.000000	100.000000	99.210526
s48	99.444444	100.000000	100.000000	99.736842
s49	98.888889	91.000000	98.000000	96.578947
s50	99.444444	81.000000	100.000000	94.736842
s51	100.000000	13.000000	100.000000	77.105263
s53	98.888889	98.000000	100.000000	98.947368
s54	94.444444	100.000000	94.000000	95.789474

Continued on next page

Table A.0.2: Between-paradigm classification accuracies for each subject and paradigm after leave-one-out training

Subject	ERP	MI	SSVEP	Total
mean	98.464052	93.588235	99.725490	97.512900
stdev	1.963687	12.691168	0.909178	3.395880

Table A.0.3: Within-paradigm classification accuracies for each subject using all train data.

Subject	ERP	MI	SSVEP	Total
s1	97.222222	61.000000	95.000000	87.105263
s2	100.000000	52.000000	100.000000	87.368421
s3	100.000000	64.000000	87.000000	87.105263
s4	97.777778	75.000000	100.000000	92.368421
s5	95.555556	55.000000	98.000000	85.526316
s7	98.888889	87.000000	100.000000	96.052632
s8	96.111111	73.000000	84.000000	86.842105
s9	100.000000	49.000000	96.000000	85.526316
s10	99.444444	54.000000	95.000000	86.315789
s11	100.000000	64.000000	100.000000	90.526316
s12	100.000000	72.000000	100.000000	92.631579
s13	100.000000	50.000000	98.000000	86.315789
s14	99.444444	50.000000	100.000000	86.578947
s15	100.000000	86.000000	91.000000	93.947368
s16	98.888889	54.000000	98.000000	86.842105
s17	95.000000	50.000000	92.000000	82.368421

Continued on next page

Table A.0.3: Within-paradigm classification accuracies for each subject using all train data.

Subject	ERP	MI	SSVEP	Total
s18	93.888889	65.000000	99.000000	87.631579
s19	100.000000	63.000000	99.000000	90.000000
s20	100.000000	63.000000	99.000000	90.000000
s21	100.000000	54.000000	100.000000	87.894737
s22	100.000000	51.000000	98.000000	86.578947
s23	80.555556	54.000000	46.000000	64.473684
s24	97.777778	59.000000	92.000000	86.052632
s25	98.888889	56.000000	100.000000	87.894737
s26	100.000000	61.000000	97.000000	88.947368
s27	100.000000	65.000000	100.000000	90.789474
s28	94.444444	91.000000	96.000000	93.947368
s29	93.888889	84.000000	96.000000	91.842105
s30	100.000000	69.000000	100.000000	91.842105
s31	97.222222	64.000000	99.000000	88.947368
s32	98.888889	56.000000	94.000000	86.315789
s33	100.000000	64.000000	94.000000	88.947368
s34	96.666667	48.000000	87.000000	81.315789
s35	99.444444	99.000000	100.000000	99.473684
s36	92.222222	73.000000	100.000000	89.210526
s37	97.222222	49.000000	98.000000	84.736842
s38	97.777778	59.000000	100.000000	88.157895
s39	97.222222	59.000000	96.000000	86.842105
s40	98.333333	57.000000	83.000000	83.421053
s41	98.333333	49.000000	97.000000	85.000000

Continued on next page

Table A.0.3: Within-paradigm classification accuracies for each subject using all train data.

Subject	ERP	MI	SSVEP	Total
s42	98.333333	61.000000	94.000000	87.368421
s43	99.444444	64.000000	100.000000	90.263158
s44	100.000000	92.000000	100.000000	97.894737
s45	99.444444	61.000000	96.000000	88.421053
s46	99.444444	76.000000	97.000000	92.631579
s48	92.777778	83.000000	100.000000	92.105263
s49	97.777778	55.000000	98.000000	86.578947
s50	100.000000	50.000000	100.000000	86.842105
s51	100.000000	9.000000	93.000000	74.210526
s53	95.000000	49.000000	94.000000	82.631579
s54	78.333333	54.000000	90.000000	75.000000
mean	97.483660	62.000000	95.411765	87.600619
stdev	4.211420	14.723597	8.179785	5.590600

Table A.0.4: Between-paradigm classification accuracies for each subject using all train data.

Subject	ERP	MI	SSVEP	Total
s1	97.222222	99.000000	100.000000	98.421053
s2	100.000000	96.000000	100.000000	98.947368
s3	100.000000	99.000000	100.000000	99.736842
s4	97.777778	99.000000	100.000000	98.684211
s5	95.555556	98.000000	99.000000	97.105263

Continued on next page

Table A.0.4: Between-paradigm classification accuracies for each subject using all train data.

Subject	ERP	MI	SSVEP	Total
s7	98.888889	98.000000	100.000000	98.947368
s8	96.111111	100.000000	100.000000	98.157895
s9	100.000000	93.000000	100.000000	98.157895
s10	99.444444	100.000000	100.000000	99.736842
s11	100.000000	100.000000	100.000000	100.000000
s12	100.000000	100.000000	100.000000	100.000000
s13	100.000000	92.000000	100.000000	97.894737
s14	99.444444	99.000000	100.000000	99.473684
s15	100.000000	99.000000	95.000000	98.421053
s16	98.888889	95.000000	98.000000	97.631579
s17	95.000000	95.000000	100.000000	96.315789
s18	93.888889	97.000000	100.000000	96.315789
s19	100.000000	94.000000	100.000000	98.421053
s20	100.000000	97.000000	100.000000	99.210526
s21	100.000000	95.000000	100.000000	98.684211
s22	100.000000	100.000000	100.000000	100.000000
s23	80.555556	91.000000	98.000000	87.894737
s24	97.777778	98.000000	95.000000	97.105263
s25	98.888889	97.000000	100.000000	98.684211
s26	100.000000	100.000000	99.000000	99.736842
s27	100.000000	100.000000	100.000000	100.000000
s28	94.444444	100.000000	99.000000	97.105263
s29	93.888889	92.000000	100.000000	95.000000
s30	100.000000	95.000000	100.000000	98.684211

Continued on next page

Table A.0.4: Between-paradigm classification accuracies for each subject using all train data.

Subject	ERP	MI	SSVEP	Total
s31	97.222222	99.000000	100.000000	98.421053
s32	98.888889	91.000000	100.000000	97.105263
s33	100.000000	97.000000	100.000000	99.210526
s34	96.666667	92.000000	99.000000	96.052632
s35	99.444444	100.000000	100.000000	99.736842
s36	92.222222	100.000000	100.000000	96.315789
s37	97.222222	98.000000	100.000000	98.157895
s38	97.777778	99.000000	100.000000	98.684211
s39	97.222222	99.000000	99.000000	98.157895
s40	98.333333	98.000000	99.000000	98.421053
s41	98.333333	97.000000	100.000000	98.421053
s42	98.333333	94.000000	100.000000	97.631579
s43	99.444444	100.000000	100.000000	99.736842
s44	100.000000	100.000000	100.000000	100.000000
s45	99.444444	99.000000	100.000000	99.473684
s46	99.444444	99.000000	100.000000	99.473684
s48	92.777778	100.000000	100.000000	96.578947
s49	97.777778	91.000000	98.000000	96.052632
s50	100.000000	88.000000	100.000000	96.842105
s51	100.000000	16.000000	99.000000	77.631579
s53	95.000000	99.000000	100.000000	97.368421
s54	78.333333	100.000000	98.000000	89.210526
mean	97.483660	95.568627	99.509804	97.512900
stdev	4.211420	11.670736	1.091366	3.609110

Bibliography

[1] Han Yuan and Bin He. Brain–computer interfaces using sensorimotor rhythms: Current state and future perspectives. *IEEE Transactions on Biomedical Engineering*, 61(5):1425–1435, 2014.

[2] Siamac Fazli, Cristian Grozea, Márton Danóczy, Benjamin Blankertz, Florin Popescu, and Klaus-Robert Müller. Subject independent EEG-based BCI decoding. In *Advances in Neural Information Processing Systems*, pages 513–521, 2009.

[3] Terence W Picton et al. The P300 wave of the human event-related potential. *Journal of clinical neurophysiology*, 9:456–456, 1992.

[4] Boyla Mainsah, Dmitry Kalika, Leslie Collins, Siyuan Liu, and Chandra Throckmorton. Information-based adaptive stimulus selection to optimize communication efficiency in brain-computer interfaces. In *Advances in Neural Information Processing Systems*, pages 4820–4830, 2018.

[5] Gernot R Müller-Putz, Reinhold Scherer, Christian Brauneis, and Gert Pfurtscheller. Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *Journal of Neural Engineering*, 2(4):123, 2005.

[6] Benjamin Blankertz, Steven Lemm, Matthias Treder, Stefan Haufe, and Klaus-Robert Müller. Single-trial analysis and classification of ERP components: a tutorial. *NeuroImage*, 56(2):814–825, 2011.

[7] Herbert Ramoser, Johannes Muller-Gerking, and Gert Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4):441–446, 2000.

[8] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2390–2397. IEEE, 2008.

[9] Steven Lemm, Benjamin Blankertz, Gabriel Curio, and K-R Muller. Spatio-spectral filters for improving the classification of single trial EEG. *IEEE Transactions on Biomedical Engineering*, 52(9):1541–1548, 2005.

[10] Min-Ho Lee, O-Yeon Kwon, Yong-Jeong Kim, Hong-Kyung Kim, Young-Eun Lee, John Williamson, Siamac Fazli, and Seong-Whan Lee. EEG dataset and OpenBMI toolbox for three BCI paradigms: an investigation into BCI illiteracy. *GigaScience*, 8(5), January 2019.

[11] Zhonglin Lin, Changshui Zhang, Wei Wu, and Xiaorong Gao. Frequency recognition based on canonical correlation analysis for ssvep-based BCIs. *IEEE Transactions on Biomedical Engineering*, 54(6):1172–1176, 2007.

[12] Albina Li, Kanat Alimanov, Siamac Fazli, and Min-Ho Lee. Towards paradigm-independent brain computer interfaces. In *2020 8th International Winter Conference on Brain-Computer Interface (BCI)*, pages 1–6. IEEE, 2020.

[13] Min Hye Chang, Jeong Su Lee, Jeong Heo, and Kwang Suk Park. Eliciting dual-frequency SSVEP using a hybrid SSVEP-p300 BCI. *Journal of Neuroscience Methods*, 258:104–113, January 2016.

[14] Feng Duan, Dongxue Lin, Wenyu Li, and Zhao Zhang. Design of a multimodal EEG-based hybrid BCI system with visual servo module. *IEEE Transactions on Autonomous Mental Development*, 7(4):332–341, December 2015.

[15] Zhihua Wang, Yang Yu, Ming Xu, Yadong Liu, Erwei Yin, and Zongtan Zhou. Towards a hybrid BCI gaming paradigm based on motor imagery and SSVEP. *International Journal of Human–Computer Interaction*, 35(3):197–205, March 2018.

[16] R Leeb, H Sagha, R Chavarriaga, and J del R Millan. Multimodal fusion of muscle and brain signals for a hybrid-BCI. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, August 2010.

[17] Yuanqing Li, Jiahui Pan, Fei Wang, and Zhuliang Yu. A hybrid BCI system combining p300 and SSVEP and its application to wheelchair control. *IEEE Transactions on Biomedical Engineering*, 60(11):3156–3166, November 2013.

[18] O-Yeon Kwon, Min-Ho Lee, Cuntai Guan, and Seong-Whan Lee. Subject-independent brain-computer interfaces based on deep convolutional neural networks. *IEEE transactions on neural networks and learning systems*, 2019.

[19] Heung-Il Suk and Seong-Whan Lee. A novel bayesian framework for discriminative feature extraction in brain-computer interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):286–299, 2013.

[20] Fabien Lotte and Cuntai Guan. Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Transactions on biomedical Engineering*, 58(2):355–362, 2010.

[21] No-Sang Kwak, Klaus-Robert Müller, and Seong-Whan Lee. A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PloS one*, 12(2):e0172578, 2017.

[22] Masaki Nakanishi, Yijun Wang, Yu-Te Wang, and Tzyy-Ping Jung. A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials. *PloS one*, 10(10), 2015.

[23] Zhicheng Yan, Hao Zhang, Robinson Piramuthu, Vignesh Jagadeesh, Dennis DeCoste, Wei Di, and Yizhou Yu. HD-CNN: hierarchical deep convolutional neural networks for large scale visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2740–2748, 2015.

[24] Thorir Mar Ingolfsson, Michael Hersche, Xiaying Wang, Nobuaki Kobayashi, Lukas Cavigelli, and Luca Benini. Eeg-tcnet: An accurate temporal convolutional network for embedded motor-imagery brain-machine interfaces, 2020.

[25] Xiangmin Lun, Shuyue Jia, Yimin Hou, Yan Shi, Yang Li, Hanrui Yang, Shu Zhang, and Jinglei Lv. Gcns-net: A graph convolutional neural network approach for decoding time-resolved eeg motor imagery signals, 2020.

[26] Yimin Hou, Lu Zhou, Shuyue Jia, and Xiangmin Lun. A novel approach of decoding EEG four-class motor imagery tasks via scout ESI and CNN. *Journal of Neural Engineering*, 17(1):016048, February 2020.