Technical University of Denmark

Principles of Brain-Computer Interface 22053

Project: SSVEP Classification Using Train-free and Deep Learning Methods

17 / 01 / 2024

Ioanna Gemou, Nika Regginou

Abstract

This report presents a comprehensive study on the application of Steady-State Visual Evoked Potentials (SSVEPs) in Brain-Computer Interface (BCI) systems, exploring a range of methodologies for signal classification. We delve into traditional train-free approaches such as Canonical Correlation Analysis (CCA) and its variants, alongside advanced deep learning techniques including a custom-made Convolutional Neural Network (CNN) and an EEGNet. The study employs a dataset encompassing SSVEP recordings from 35 subjects, focusing on both method development and performance evaluation. We compare the efficacy of traditional and modern classification methods, assessing their accuracy and Information Transfer Rate (ITR) across different subjects. Notably, we observed consistent accuracy rates across a diverse group of subjects, highlighting the robustness of these methods to individual variability. We also identify potential areas for future work, including the need for model personalization and optimization in deep learning approaches.

Contents

1 Introduction

Brain Computer Interfaces (BCIs) are a tool for people to communicate with a computer or external device only by using their thoughts. BCI is defined as a system that captures brain signals, analyses them, and gives an output in the form of a command to control an external device (figure 1) [1]. This can prove to be very useful in cases were the subject cannot communicate with their surroundings or move as they wish due to a disability or impairment, like those with locked-in syndrome [2]. Historically BCIs were used mainly for the purpose of assisting people with disabilities and to improve their quality of life, mainly in the areas of assistive technology and neurohabilitation (figure 1) [3]. But BCIs are not only limited to that. They can enhance brain-computer interactions for more effective and efficient outputs, for example in the gaming industry or the military [3].

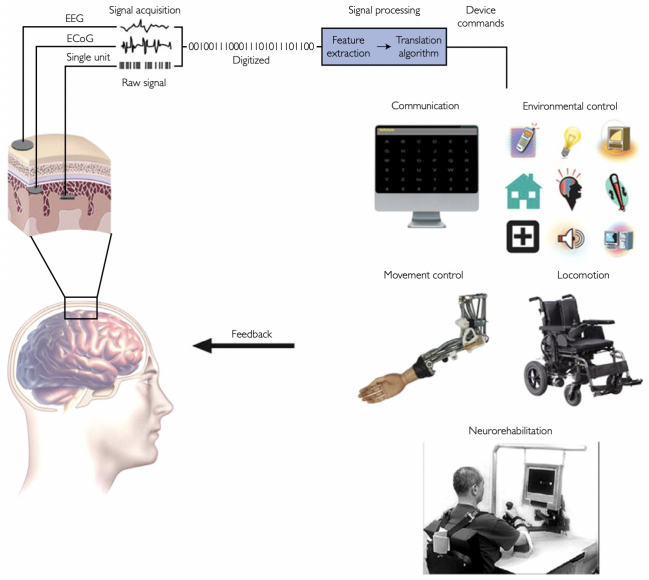

Figure 1: Brain computer interface (BCI) block diagram. Brain waves are captured using various methods, mainly differing on incisiveness or how close to the brain are the electrodes positioned. These brain waves are analysed using signal processing techniques and translated into commands for external devices. These devices can range from spelling software on a computer, smart home control, prosthesis manipulation and wheelchair driving. BCIs can also be used for neurohabilitation, to create an interactive environment to enhance the effects of motor imagery used in physiotherapy [1].

Despite many advances of BCI systems over the years, this field is still in its infancy [4]. The main bottleneck for online applications is often the rate of communication [4]. Therefore a method with high-speed communication is needed. SSVEP or steady state visually evoked potentials are often used in application were high communications rates are needed. Some of the highest transfer rates in BCIs were achived in system utilizing SSVEPs [5].

The purpose of our project was to investigate various signal processing methods for SSVEP detection in offline EEG signals.

1.1 SSVEPs for BCIs

SSVEPs are widely used in BCI applications due to their simplicity and capacity to reach high information transfer rates (ITRs) [4, 5]. To understand why these potentials can achieve such ITRs, we need to understand the biology behind SSVEPs.
As mentioned, SSVEP stands for steady-state visually evoked potentials. Visually evoked means, potentials that are generated when the subject is focusing on a visual stimulus [5]. Since these EPs are visually evoked, they are the response of populations of neurons in the visual cortex [5]. Therefore, brain signals are collected from the visual cortex in the occipital lobe (figure 2). Following the most commonly used acquisition method, electroencephalography (EEG) [4], signals are captured from the back of the scalp. According to the extended 10-20 EEG electrode positioning system. The location of the electrodes that has been shown to have the highest amplitude of SSVEP signals are annotated in red in figure 2.

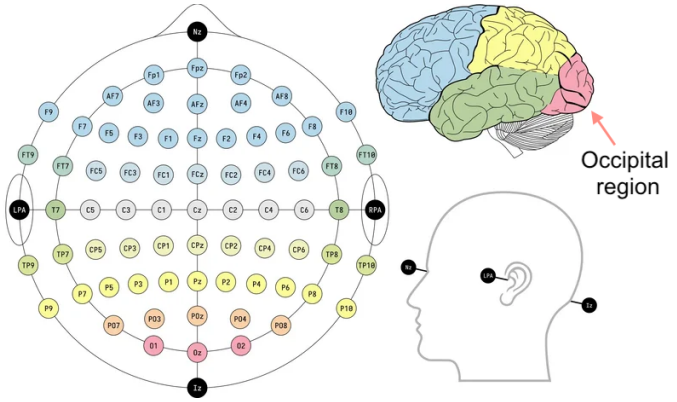

Figure 2: Electroncephalography (EEG) 10-20 system electrode placement. In the image on the top right we can see electrode positions colour coded with areas of the brain which they correspond to.
(Image source: `https://commons.wikimedia.org/wiki/File:EEG_10-10_system_with_additional_information.svg`)

SSVEPs are generated when the subject is focusing on a flickering pattern. These EPs appear approximately 300ms after stimulus onset [6]. The frequency of the flickering patters appears as the frequency of the brain waves captured on the frequency spectrum along with other peaks at the harmonic frequencies [4, 5] (figure 3). This is due to the non-linearity of the brain as a system. The frequency can range from 1 to 100 Hz [6].

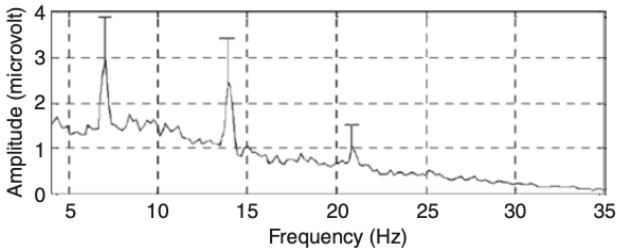

Figure 3: Example of a mean power spectrum of SSVEPs calculated by fast-Fourier transform (FFT). Prominent peaks can be seen at around 7 Hz, 14 Hz etc corresponding to the stimulus frequency of 7 Hz and its harmonics [5].

Subsequently, if different stimuli are associated with different frequencies, then a BCI system can decode the stimulus from the EEG [5]. In other words, it will be possible to determine on which visual stimuli the subject was focusing on at the given time only by looking at the EEG signals, without the use of eye-tracking or other similar methods.
The flickering patterns are often generated on a computer screen which the subject focuses on and uses to control the external device (figure 4). Since these EPs are visually evoked, no previous training is needed for them to be used in a BCI application, unlike motor imagery [5]. It must also be noted that SSVEPs are less susceptible to artifacts caused by eye-blinking or other electromyographic noise [6], mainly due to the location of the brain region in which they are generated in. The above reasons make SSVEPs a more reliable and simple approach to designing BCI systems.
Common applications of BCIs utilizing SSVEPs are visual spellers, the user can spell out words and sentences by focusing their gaze on a particular letter on a matrix [5]. Other applications include locomotion and home appliances control.

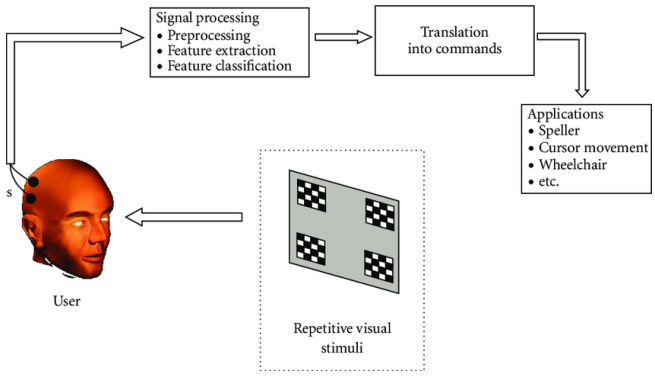

Figure 4: Block diagram of a BCI system utilizing SSVEPs [7]

1.2 Related Work

The earliest accounts of SSVEP related work, are by D. Regan in 1995 [8]. He experimented with long trains of sinusoidal monochromatic light stimulus. It was observed that these stimuli cause a stable VEP. By averaging this VEP over multiple trials, he was able to extract VEPs of reasonable amplitude. Since the stimulus was a train of sinusoidal light, these EEG patterns were termed "steady-state" visually evoked potentials, or SSVEP. Since then SSVEPs have started to emerge in BCI applications making a very popular approach in the field today.

Multichannel frequency recognition methods are common in BCI systems utilizing SSVEPs. The work by [9] proposed a novel multi-variate synchronization index (MSI) for frequency recognition. This metric determined the level of synchronization between multichannel EEGs and the reference signals. It is based on the S-estimator as the index. The S-estimator relies on the entropy derived from the normalized eigenvalues of the correlation matrix of multivariate signals. The researchers concluded, that this method performs better than the traditional CCA method, especially of short length recordings and fewer channels [9].

A different work by [10] proposed a Hilbert-Huang transform (HHT) based method for SSVEP detection. The HHT is a time-frequency based transform. An eigenvector as the outcome of HHT was regarded as the distinctive feature capturing various frequency components of the SSVEP signal. A Fisher classifier was used for classification with the eigenvector as its input. Compared to the traditional fast-Fourier transform, this method can reach classification accuracies of more than 85%.

2 Methods

In this work we used various signal processing methods and deep learning algorithms to analyse and classify EEG signals to their corresponding SSVEP frequencies. These methods were applied on offline EEG waves. The dataset used, was derived from [11]. This is a benchmark dataset for SSVEP-based Brain computer interfaces. The purpose of this work was to investigate a handful of methods for SSVEP detection.

2.1 Dataset

The open dataset by [11] was created to serve as a benchmark dataset for more adequate algorithm comparison between research groups.
The dataset comprised of EEG SSVEP signals acquired with a visual speller with 40 targets. The target matrix can be seen in figure 5.

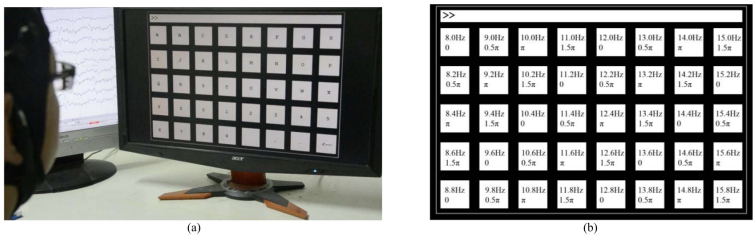

Figure 5: (a) Subject and monitor set-up for EEG acquisition (b) Target matrix used in data acquisition of SSVEPs with corresponding flickering frequency and phase [11]

A total of 35 healthy subjects participated in the EEG recording. The subjects were prompted to focus their gaze at a specific character on the monitor with a red mark. The recordings were broken down to 6s for each trial, 0.5s pre-stimulus and 5.5s post-stimulus onset. Six trials per character were recorded. This totals to 40×6 EEG recordings per subject, 40 characters or frequencies $\times$ 6 trials. Each character was prompted at random during the recording process. The EEG signals were recorded with the Synamps2 EEG system (Neuroscan, Inc.) at 1000 Hz and later downsampled to 250 Hz to reduce computational load. The bandwidth of the EEG signals ranged from 0.15 Hz to 200 Hz.
A total of 64 electrodes were used with locations according to the extended 10-20 system. The reference electrode was placed at Cz. The electrode impedance was kept under $10k\Omega$.

2.2 Data Loading & Preprocessing

For the unsupervised method Canonical Correlation Analysis with Power spectrum as input (CCA-based PS analysis) (see description 2.3.2), MATLAB version R2021b was used. For the rest of the algorithms, python in GoogleColab was used.
The signals in the dataset were already filtered at 50 Hz with a notch filter. A bandpass filter ranging from 5 Hz to 100 Hz was applied to all EEG signals for CCA-based PS analysis and a 7 Hz to 90 Hz for the rest of the algorithms.
Taking into account the biology behind SSVEP, the most appropriate channels to consider are PO5, PO3, POz, PO4, PO6, Pz, O1, Oz, and O2 (see figure 2 in previous section on SSVEPs 1.1). For some of the methods (primarily in section 2.3.2) a reduction in computational load was needed, therefore, from a manual inspection of some EEG example from the dataset, it was concluded that a single channel from those the ones above would be sufficient since all three channels exhibited similar EEG patterns and power spectrum (figure 6). The channel selected was Oz.

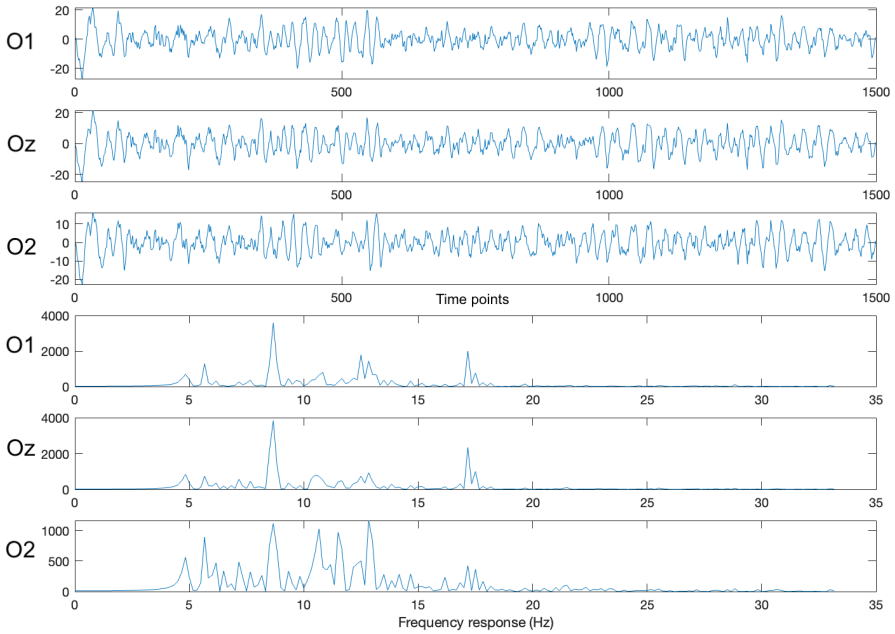

Figure 6: Visual inspection of an EEG example from the dataset, a single channel would be sufficient since all three channels exhibited similar EEG patterns and power spectrum.

We employed Independent Component Analysis (ICA) for artifact removal, such as eye blinking. ICA is a powerful technique for isolating and removing artifacts from EEG signals by separating the mixed signals into independent components. However, it should be mentioned that the ICA did not specifically identify any components as artifacts related to eye blinking. This observation is consistent with the experimental protocol where subjects were instructed to avoid blinking during the recordings. Not to mention that the signals originate from electrodes on the back of the scalp making them less susceptible to electromyographic noise [6]. Nonetheless, the use of ICA was crucial in ensuring the overall cleanliness of the data by allowing for the identification and removal of other potential artifacts.

2.3 Train-free or Unsupervised methods

Three unsupervised methods were used to classify SSVEP frequencies from the EEG signals. Standard canonical correlation analysis (Standard-CCA), canonical correlation analysis with power spectrum as input (CCA-based PS analysis) and Filter-Bank canonical correlation analysis (FBCCA). All of which are CCA-based.

2.3.1 Canonical Correlation Analysis

Canonical correlation analysis (CCA) has been used widely for SSVEP detection [12]. This method calculates the correlation between two variables [12]. In this case the input signal $\mathbf{X}$ and a reference

variable $\mathbf{Y}$. Considering their linear combinations $\mathbf{x} = \mathbf{X^T W_X}$ and $\mathbf{y} = \mathbf{Y^T W_Y}$, CCA computes the weights $\mathbf{W_X}$ and $\mathbf{W_Y}$ so that the correlation between $\mathbf{X}$ and $\mathbf{Y}$ is maximized. The objective function of the correlation is calculated as described in equation 1.

$$\max_{\mathbf{W_X},\mathbf{W_Y}} \rho(\mathbf{x},\mathbf{y}) = \frac{E(\mathbf{W_X^T X Y^T W_Y})}{\sqrt{E(\mathbf{W_X^T X X W_X})E(\mathbf{W_Y^T Y Y^T W_Y})}} \tag{1}$$

The input signal are the EEG waves in time domain, and reference signals are sinusoidal waves with frequency equal to the target frequencies and their harmonics as described in equation 2.

$$\mathbf{Y_f} = \begin{bmatrix} sin(2\pi ft) \\ cos(2\pi ft) \\ \vdots \\ sin(2\pi N_h ft) \\ cos(2\pi N_h ft) \end{bmatrix} \tag{2}$$

Were f is target frequency, and N_h number of harmonics. The frequency corresponding to the reference signals exhibiting the highest correlation is regarded as the frequency of SSVEPs.

2.3.2 Canonical Correlation Analysis with Power Spectrum as Input

This method is very similar to Standard-CCA differing in the type of variables compared. Instead of $\mathbf{X}$ as EEG signals in time domain, the Power spectrum of that signal is calculated using the fast-Fourier transform algorithm. The correlation ρ values were calculated using the canoncorr() function in MATLAB (refer to Appendix A for full MATLAB code). $\mathbf{Y}$ is the Power spectrum of the reference frequencies, which is equal to a dirac function. The Y_f vector only consists of cosine waves since the power spectra of a sine wave and a cosine wave are identical (equation 3).

$$\mathbf{Y_f} = \begin{bmatrix} \mathcal{F}\{cos(2\pi ft)\} \\ \vdots \\ \mathcal{F}\{cos(2\pi N_h ft)\} \end{bmatrix} = \begin{bmatrix} \delta(f) \\ \vdots \\ \delta(N_h f) \end{bmatrix} \tag{3}$$

2.3.3 Filter-Bank Canonical Correlation Analysis

Filter-bank Canonical correlation analysis is an extension of standard CCA. In FBCCA instead of calculating the correlation of the raw EEG signal to the reference sinusoidal, the EEG signal is filtered in sub-bands corresponding to the target frequency. A flowchart of the algorithm can be seen in figure 7.

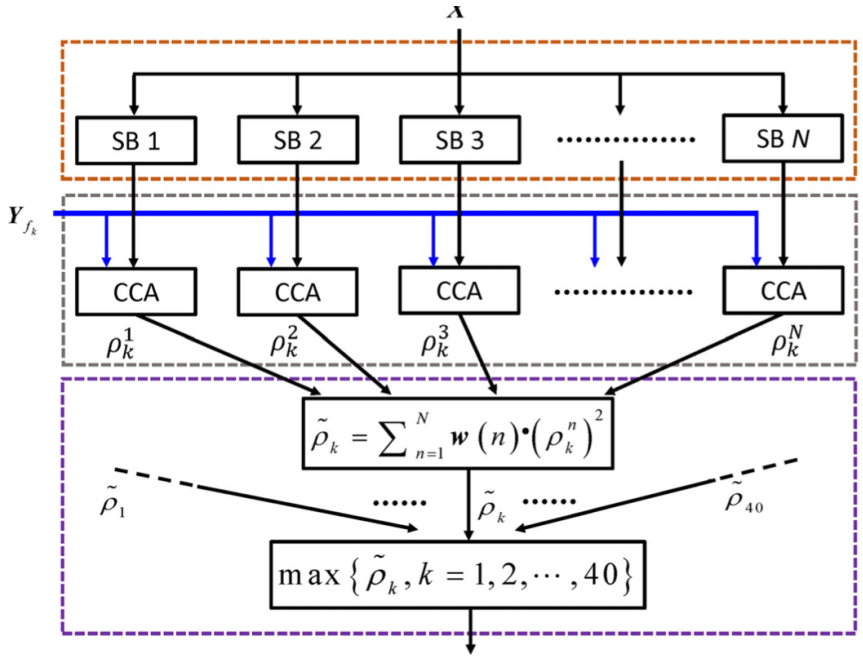

Figure 7: Flowchart of the Filter-Bank Canonical Correlation Analysis algorithm [12]

The FBCCA algorithm consists of three major processes. First the filter bank analysis, secondly, CCA between the sub-bands and the reference signals, and lastly the SSVEP target identification. In this work, the filtfilt() method was used in python to filter the signal and create the sub-bands. This method applies a a Chebyshev type I infinite impulse response (IIR) filter. The bandwidth of the sub-bands was selected to be from 8 Hz to $80 \times N$ Hz with equal steps of 8 Hz in-between. N denotes the number of sub-bands (figure 8).

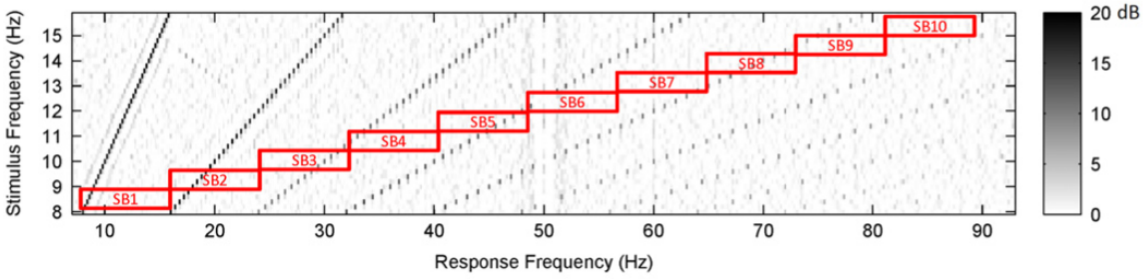

Figure 8: The frrequency response bandwidth of [8 Hz 88 Hz] divided into 10 sub-bands, each of which has a bandwidth of 8 Hz [12]

Following this procedure, standard CCA is applied. To classify the target frequencies, the maximum correlation is calculated, and the frequency corresponding to that is denoted as the frequency of the

SSVEP.

2.4 Deep Learning

2.4.1 Convolutional Neural Network

A convolutional neural network (CNN) is developed for analyzing EEG data, using TensorFlow [13] and Keras [14]. The model follows a sequential layout with two convolutional layers. Each layer is followed by batch normalization, ReLU activation, max pooling, and dropout layers to improve the model's accuracy and prevent overfitting. The first layer has 32 filters and the second has 64 filters, both with a size of 3x3. The architecture also includes flattening and fully connected layers, ending with a softmax output layer for classifying into 40 different categories.

Layer (type)	Output Shape	Param #
conv2d_25 (Conv2D)	(None, 6, 1498, 32)	320
batch_normalization_36 (BatchNormalization)	(None, 6, 1498, 32)	128
activation_36 (Activation)	(None, 6, 1498, 32)	0
max_pooling2d_24 (MaxPooling2D)	(None, 3, 749, 32)	0
dropout_36 (Dropout)	(None, 3, 749, 32)	0
conv2d_26 (Conv2D)	(None, 1, 747, 64)	18496
batch_normalization_37 (BatchNormalization)	(None, 1, 747, 64)	256
activation_37 (Activation)	(None, 1, 747, 64)	0
max_pooling2d_25 (MaxPooling2D)	(None, 1, 747, 64)	0
dropout_37 (Dropout)	(None, 1, 747, 64)	0
flatten_12 (Flatten)	(None, 47808)	0
dense_24 (Dense)	(None, 128)	6119552
batch_normalization_38 (BatchNormalization)	(None, 128)	512
activation_38 (Activation)	(None, 128)	0
dropout_38 (Dropout)	(None, 128)	0
dense_25 (Dense)	(None, 40)	5160

Total params: 6144424 (23.44 MB)
Trainable params: 6143976 (23.44 MB)
Non-trainable params: 448 (1.75 KB)

Figure 9: The components of the CNN model

The model is evaluated using a **Leave-One-Subject-Out (LOSO)** cross-validation method [15], a specific form of k-fold cross-validation where k is equal to the number of samples in the dataset. This approach involves training the model on data from all but one subject and testing it on the remaining subject's data. This is repeated so that each subject's data is used as a test set once, ensuring a comprehensive assessment of the model's ability to generalize to new subjects.

The training process is enhanced with the Early Stopping method, to stop training if there's no im-

provement in validation loss and a Learning Rate Scheduler, to adjust the learning rate based on the validation loss. The model's performance is then evaluated on the test data from the excluded subject, providing a measure of its accuracy across different subjects.

2.4.2 EEGNet

EEGNet is a compact convolutional neural network architecture specifically designed for decoding and analyzing electroencephalography (EEG) signals. It was developed by Lawhern et all [16], and was first introduced in a paper published in 2018. The primary goal of EEGNet is to facilitate EEG-based brain-computer interface (BCI) applications by providing an efficient and effective means of interpreting EEG data. It is particularly known for its small model size and its ability to work well across different BCI paradigms, including event-related potentials (ERPs), sensory-motor rhythms (SMRs), and movement-related cortical potentials (MRCPs).

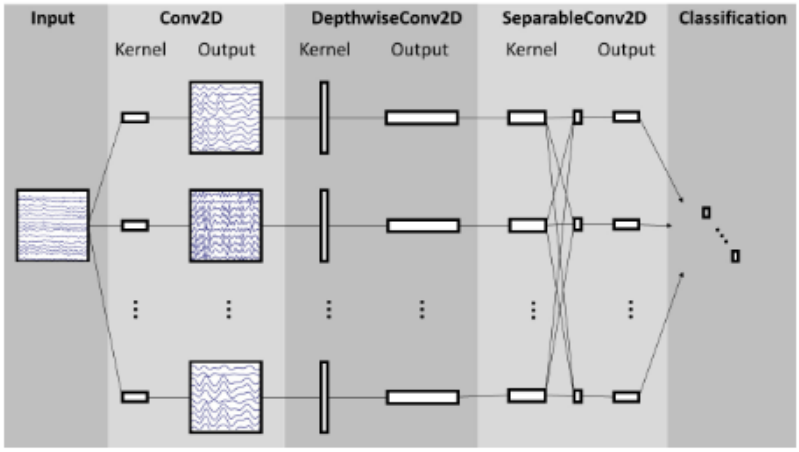

Figure 10: EGGNet Architecture, as proposed by Lawhern et all in 2018 [16]. The network initiates with a temporal convolution (illustrated in the second column) that is designed to extract frequency filters. This is succeeded by a depthwise convolution (depicted in the middle column), which is applied to each individual feature map, enabling the network to learn spatial filters that are specific to different frequencies. The separable convolution (shown in the fourth column) combines two convolutional processes: a depthwise convolution that provides a temporal summary for each feature map on an individual basis, and a pointwise convolution that is responsible for optimally integrating these feature maps together.

In this work, the EEGNet architecture was employed in combination with Leave-One-Subject-Out (LOSO) cross-validation.

2.5 Performance Evaluation

In the classification results section of the study, two key metrics were utilized to assess the performance of the model: Information Transfer Rate (ITR) [17] and classification accuracy.

The **ITR** provides a measure of how effectively information is conveyed in a BCI system, typically expressed in bits per minute. It considers the number of target options in the system, the accuracy of selection, and the time required for each selection. The ITR calculation computes the metric based on the formula involving logarithmic terms of the number of targets and accuracy, adjusted for selection time.

$$ITR = \begin{cases} \frac{60}{T}\log_2(N), & \text{if } p = 1 \\ \frac{60}{T}\left[\log_2(N) + p\log_2(p) + (1-p)\log_2\left(\frac{1-p}{N-1}\right)\right], & \text{otherwise} \end{cases} \tag{4}$$

where N is the number of targets, p is the accuracy (proportion of correct predictions), and T is the average selection time in seconds. In this work, the value of 5 sec was considered for T, since according to the dataset description [12], the subjects had 5 seconds to shift their gaze on to the target. Since we don't know the decision making time of each subject, we considered this safe assumption. The expression provides the value of ITR in bits per minute.

Accuracy is a fundamental metric in classification tasks. This function compares the ground truth labels with the predicted labels to determine the proportion of correct predictions. Accuracy is an essential metric as it directly reflects the model's ability to correctly classify EEG signals into the intended categories.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{5}$$

3 Results

3.1 Train-free or Unsupervised methods

3.1.1 Standard CCA

In our study, we observed notable variations in classification accuracy and Information Transfer Rate (ITR) across the 35 subjects. The accuracy percentages ranged from as low as 37.84% to a perfect 100%, while ITR values spanned from 28.81 bits/min to a maximum of 63.86 bits/min, reflecting the diversity in cognitive performance among the participants.

Table 1: Classification Accuracy and ITR for 35 Subjects

Subject	**Accuracy (%)**	**ITR (bits/min)**
1	92.47	54.46
2	96.65	59.20
3	100.00	63.86
4	99.58	63.13
5	100.00	63.86
6	100.00	63.86
7	98.33	61.33
8	88.70	50.59
9	84.52	46.58
10	99.16	62.50
11	75.73	38.88
12	99.58	63.13
13	94.56	56.76
14	100.00	63.86
15	99.16	62.50
16	66.11	31.28
17	98.33	61.33
18	85.36	47.36
19	74.48	37.84
20	99.16	62.50
21	80.75	43.18
22	96.23	58.70
23	94.56	56.76
24	98.74	61.90
25	99.58	63.13
26	100.00	63.86
27	98.74	61.90
28	91.21	53.14
29	75.31	38.53
30	98.33	61.33
31	100.00	63.86
32	100.00	63.86
33	62.76	28.81
34	100.00	63.86
35	100.00	63.86
Average	**92.8**	**56.04**

3.1.2 CCA with PS as input

The algorithm was run on all 35 subjects. The accuracy and ITR can be seen in table 2.

Table 2: Classification Accuracy and ITR for 35 Subjects for CCA PS algorithm

Subject	**Accuracy (%)**	**ITR (bits/min)**
1	45	17.07
2	100	63.86
3	100	63.86
4	97.5	60.25
5	100	63.86
6	100	63.86
7	97.5	60.25
8	100	63.86
9	87.5	49.41
10	100	63.86
11	95	57.25
12	100	63.86
13	62.5	28.63
14	95	57.26
15	100	63.86
16	87.5	49.41
17	100	63.86
18	92.5	54.49
19	100	63.86
20	100	63.86
21	50	20.15
22	100	63.86
23	97.5	60.25
24	100	63.86
25	100	63.86
26	100	63.86
27	100	63.86
28	95	57.26
29	87.5	49.41
30	100	63.86
31	100	63.86
32	100	63.86
33	37.5	12.77
34	97.5	60.25
35	100	63.86
Average	92.14	56.3253

A total of 20 out of 35 subjects had a perfect accuracy of 100%, with only 3 subjects at a low accuracy. The rest of the subjects' accuracies range from 87.5% to 97.5%, with an exception of subject 13 at 62.5%. Subject 1, 21 and 33 had an accuracy of 45%, 50% and 33% respectively. This is mainly due to

the fact that the power spectra of these subjects were found to have artifacts around frequency 10 Hz. This caused many false identifications. In future work, the cause of these artifacts can be investigated and removed.

3.1.3 FBCCA

Due to limited time and resources the FBCCA method was tested only on five subjects.
The accuracies for the five subjects range from 92.47% to 100%, with Subjects 3 and 5 achieving perfect accuracy. Subject 1, while still performing well, had the lowest accuracy at 92.47%. The ITR values, which provide insight into how quickly a subject can convey information using the BCI, range from 54.46 bits/min for Subject 1 to 63.86 bits/min for Subjects 3 and 5. These results suggest a high level of performance and control over the BCI across all subjects.

Subject	Accuracy (%)	ITR (bits/min)
1	92.47	54.46
2	96.65	59.20
3	100.00	63.86
4	99.58	63.13
5	100.00	63.86

Table 3: FBCCA Results for 5 Subjects

3.2 Deep Learning

3.2.1 CNN

We trained the proposed Convolutional Neural Network (CNN) for 2 epochs using a batch size of 64. We employed Leave-One-Subject-Out (LOSO) cross-validation on EEG data from 35 subjects to assess the CNN's classification accuracy and its generalization performance.

Table 4: Results for 35 folds (Subjects)

Subject	**Accuracy (%)**	**Test Loss (%)**	**ITR (bits/min)**
1	92.47	27	54.94
2	96.65	34	49.41
3	100.00	35	51.89
4	99.58	10	57.74
5	100.00	12	59.73
6	100.00	3	63.13
7	98.33	15	56.31
8	88.70	4	62.50
9	84.52	23	52.32
10	99.16	24	58.22
11	75.73	49	37.93
12	99.58	19	60.79
13	94.56	16	58.22
14	100.00	70	28.33
15	99.16	7	61.91
16	66.11	32	50.23
17	98.33	4	62.50
18	85.36	16	57.74
19	74.48	47	47.81
20	99.16	50	41.43
21	80.75	50	45.11
22	96.23	13	57.74
23	94.56	81	26.84
24	98.74	21	55.39
25	99.58	14	58.72
26	100.00	19	61.91
27	98.74	7	60.25
28	91.21	15	54.94
29	75.31	37	50.23
30	98.33	37	43.25
31	100.00	1	63.13
32	100.00	0	63.86
33	62.76	1	14.16
34	100.00	6	61.91
35	100.00	17	56.78
Average	**92.99**	**26**	**54.94**

The results obtained from the classification task on EEG data using a Convolutional Neural Network (CNN) show promising performance across 35 subjects. The accuracy values for individual subjects range from 62.76% to a perfect 100.00%.

On average, the CNN model achieved an accuracy of approximately 92.99%, indicating that it can successfully classify EEG data into one of 40 classes. This level of accuracy is encouraging, as it suggests that the model has learned meaningful patterns in the EEG signals.

The test loss represents the model's error on the test dataset. Lower test loss values indicate that the model's predictions are closer to the ground truth labels. In this study, the test loss values vary across subjects, with the lowest being around 0.03 and the highest being 1. The average test loss across all subjects is approximately 0.26, which is relatively low. This suggests that the CNN model generalizes well to unseen EEG data, minimizing prediction errors during inference.

In summary, the CNN-based classification approach demonstrates competitive accuracy and ITR values for EEG data across multiple subjects. These results highlight the potential of CNNs in analyzing EEG data for various applications. Further optimization and fine-tuning of the model may lead to even better performance in the classification task.

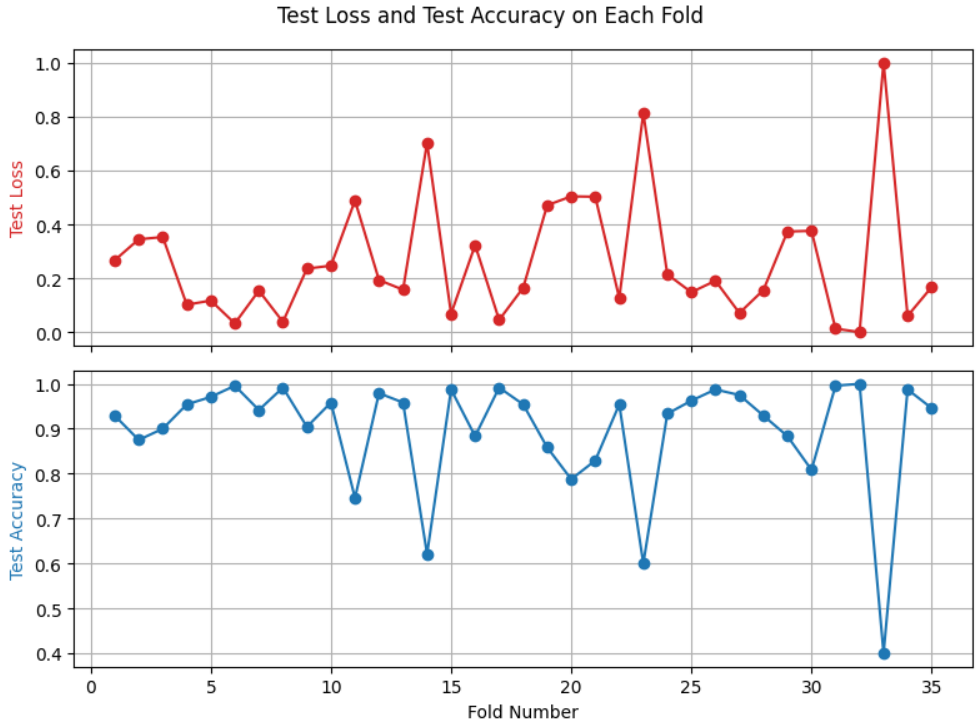

Figure 11: Test Loss and Test Accuracy on Each Fold

3.2.2 EEGNet

We trained the EEGNet for 2 epochs using a batch size of 64. We employed Leave-One-Subject-Out (LOSO) cross-validation on EEG data from 35 subjects to assess the CNN's classification accuracy and its generalization performance.

The EEGNet results show variability in performance across different subjects with an average accuracy of 55.63% and an average loss of 35.65%. This suggests that while EEGNet can classify EEG signals to a certain degree, there is significant room for improvement. The ITR varies widely from 0.02 to 60.79 bits/min, indicating inconsistent efficiency across the system. The large range in ITR values highlights the potential for EEGNet in certain scenarios but also underscores the need for optimization to ensure reliability and practical application in real-time BCI systems. Further personalization and refinement of the model could improve its robustness and effectiveness for individual users.

Fold	Accuracy	Loss (%)	ITR (bits/min)
1	21.67	63.26	5.13
2	72.50	27.82	36.24
3	55.42	37.46	23.69
4	65.00	26.98	30.46
5	72.08	33.38	35.91
6	81.67	11.59	43.99
7	41.67	39.75	15.11
8	84.58	08.45	46.64
9	77.50	23.84	40.36
10	39.58	50.06	13.92
11	53.33	35.40	22.30
12	87.08	17.57	49.01
13	60.42	31.78	27.14
14	06.25	00.00	0.35
15	79.17	17.86	41.79
16	27.50	54.13	7.70
17	86.67	11.00	48.61
18	57.08	39.62	24.82
19	03.33	81.26	0.02
20	05.00	68.85	0.17
21	26.67	59.68	7.31
22	81.67	10.51	43.99
23	30.00	49.25	8.89
24	32.92	47.56	10.35
25	38.75	31.37	13.46
26	71.25	19.12	35.24
27	79.17	16.74	41.79
28	72.08	29.70	35.91
29	35.42	44.63	11.65
30	62.92	32.94	28.93
31	87.50	04.28	49.41
32	97.92	00.00	60.79
33	04.17	88.02	0.08
34	84.17	11.47	46.26
35	65.00	22.55	30.46
Average	**55.63**	**35.65**	**26.79**

Table 5: Accuracies, Losses, and ITR for EEGnet Across All Subjects (Folds)

To enhance EEGNet's performance, focus could be placed on data quality and model refinement. Improving preprocessing of EEG signals to reduce noise and artifacts, along with implementing data augmentation, may provide more robust training inputs. Adjusting hyperparameters and experimenting

with model architecture, including layers and filters, can tailor the network to better capture the nuances of EEG data. Additionally, incorporating regularization techniques and exploring ensemble models may improve generalization. Personalizing models to individual users' EEG patterns and expanding the dataset size could further boost accuracy and reliability.

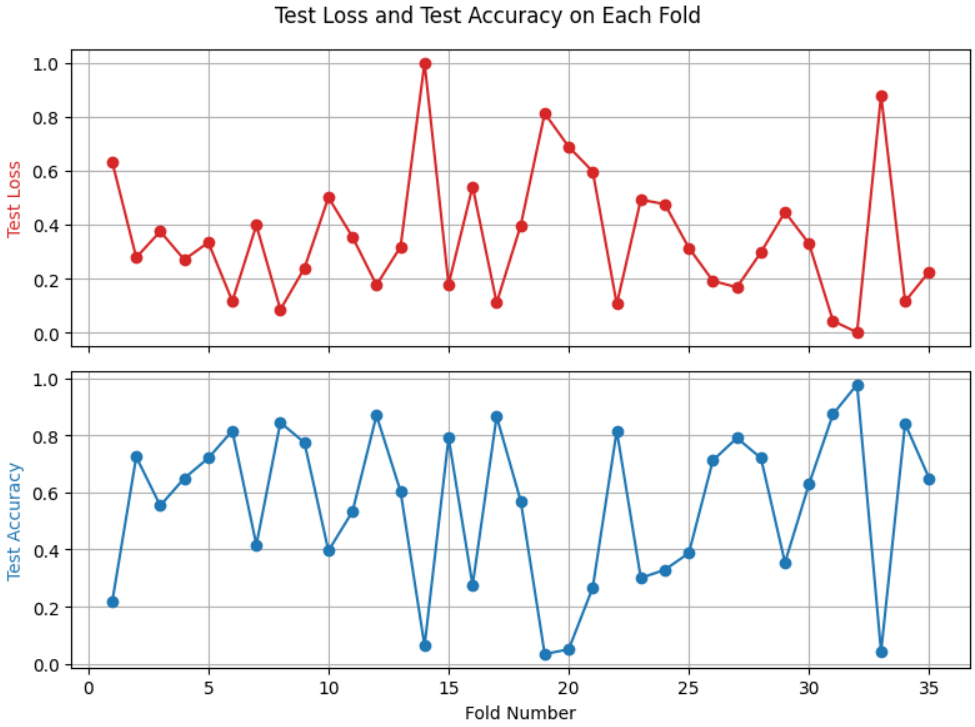

Figure 12: Test Loss and Test Accuracy on Each Fold

4 Discussion & Limitations

4.1 Discussion

It's evident that each methodology for SSVEP classification in BCIs has its unique strengths and limitations. Traditional train-free approaches like CCA and Filter-Bank CCA offer the benefit of simplicity and quick setup, as they require minimal calibration. However, our analysis indicates that these methods might not always provide the highest accuracy, especially in diverse subject groups.

On the other hand, advanced deep learning techniques, including our custom-made CNN and EEGNet, demonstrate a significant potential for higher accuracy. The CNN, in particular, showed promising results, suggesting that deep learning models can effectively learn and classify complex patterns in EEG data. Yet, the variability in performance across subjects, especially noted in EEGNet, highlights

to the need for further model refinement and personalization. This is crucial for ensuring consistent performance across a broader subjects.

One important observation from our study is the role of methodological diversity in BCI research. The comparison between traditional and modern approaches highlights that a singular approach may not fit all scenarios in BCI applications. Therefore, a hybrid approach or a combination of methodologies might be beneficial in certain contexts, including the strengths of both traditional and advanced techniques.

In addition to the comparison of methodologies, an important finding in our study is the relatively uniform accuracy across individual subjects, irrespective of the chosen classification method. Despite the diversity in the subject group, encompassing different genders, ages, and experience levels with BCIs, the accuracy rates of both traditional methods like CCA and Filter-Bank CCA, as well as advanced techniques such as CNN and EEGNet, remained more or less consistent across individuals. This uniformity in performance suggests that the methodologies employed are robust to subject variability, which is a crucial aspect in the development of universally effective BCI systems.

4.2 Limitations

Several limitations should be considered when interpreting the results. Firstly, the variability in SSVEP responses across subjects presents a significant challenge. This diversity in cognitive and physiological responses can affect the performance of both traditional and deep learning methods, making it difficult to develop a universally effective model.

Additionally, the computational complexity of deep learning models, especially in terms of training and real-time processing, poses a challenge. The requirement for substantial computational resources may limit the practical application of these models in real-time BCI systems.

It is also important to note that the train-free methods, while simpler and quicker to set up, may not provide the nuanced analysis possible with more sophisticated models. This could lead to lower accuracy or less efficient information transfer in certain scenarios.

Addressing these limitations in future research is crucial for advancing the field of BCI and developing more effective and reliable systems for a wide range of applications.

5 Conclusion

In our study, we explored various methodologies for SSVEP classification in BCIs, including traditional train-free methods like CCA and advanced deep learning approaches such as a custom-made CNN and

the EEGNet. The results indicate that custom CNN and CCA and FBCCA, generally outperformed traditional methods in terms of accuracy. The EEGNet, while effective in certain scenarios, showed a wider range of performance across subjects, suggesting a need for more targeted refinement to enhance its reliability and practical application in real-time BCI systems. Overall, the comparison between methods underlines the importance of exploring different classification methods and highlights the potential benefits of combining traditional and modern approaches in BCI research.

5.1 Further Work

We propose some alternative methodologies that may result in more effective categorization of SSVEP-based brain-computer interfaces.

- **Hybrid Modeling Approaches**: Integrating signal processing methods with deep learning architectures could lead to improved feature extraction and more accurate EEG signal interpretation.
- **Temporal Dynamics with RNNs**: Implementing Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network, could provide a more nuanced understanding of EEG temporal patterns. RNNs are particularly well-suited for handling time-series data, offering potential improvements in signal classification [18].
- **Transformer-Based Deep Neural Network Model**: Recent advancements in deep learning, particularly the Transformer architecture, present a novel approach for SSVEP classification. A Transformer-based model, such as the proposed SSVEPformer, can effectively handle the spectral and spatial information in EEG data. This method holds the potential to improve classification performance in inter-subject scenarios by leveraging the Transformer's superior feature extraction capabilities and accommodating the unique characteristics of SSVEP signals [19].
- **Data Augmentation with GANs**: Employing Generative Adversarial Networks (GANs) for creating additional EEG data could help address challenges such as overfitting and improve the model's generalization capabilities. This method can be particularly useful in scenarios with limited EEG datasets [20].

6 References

[1] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, "Brain-computer interfaces in medicine," in Mayo clinic proceedings, vol. 87, pp. 268–279, Elsevier, 2012.

[2] A. Kübler, "The history of bci: From a vision for the future to real support for personhood in people with locked-in syndrome," Neuroethics, vol. 13, no. 2, pp. 163–180, 2020.

[3] N. Hill and J. Wolpaw, "Brain–computer interface," in Reference Module in Biomedical Sciences, Elsevier, 2016.

[4] X. Chen, Y. Wang, M. Nakanishi, X. Gao, T.-P. Jung, and S. Gao, "High-speed spelling with a noninvasive brain–computer interface," Proceedings of the national academy of sciences, vol. 112, no. 44, pp. E6058–E6067, 2015.

[5] R. P. Rao, Brain-computer interfacing: an introduction. Cambridge University Press, 2013.

[6] J. Bieger and G. Garcia-Molina, Light Stimulation Properties to Influence Brain Activity: A Brain-Computer Interface Application. 09 2010.

[7] 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, vol. 2010, 2010.

[8] D. Regan, "Some characteristics of average steady-state and transient responses evoked by modulated light," Electroencephalography and clinical neurophysiology, vol. 20, no. 3, pp. 238–248, 1966.

[9] Y. Zhang, P. Xu, K. Cheng, and D. Yao, "Multivariate synchronization index for frequency recognition of ssvep-based brain–computer interface," Journal of neuroscience methods, vol. 221, pp. 32–40, 2014.

[10] L. Zhao, P. Yuan, L. Xiao, Q. Meng, D. Hu, and H. Shen, "Research on ssvep feature extraction based on hht," in 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, vol. 5, pp. 2220–2223, 2010.

[11] Y. Wang, X. Chen, X. Gao, and S. Gao, "A benchmark dataset for ssvep-based brain–computer interfaces," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 10, pp. 1746–1752, 2016.

[12] 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," Journal of neural engineering, vol. 12, no. 4, p. 046008, 2015.

[13] "TensorFlow." `https://www.tensorflow.org/`.

[14] "Keras." `https://keras.io`.

[15] D. Berrar, Cross-Validation, p. 542–545. Elsevier, 2019.

[16] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: A compact convolutional network for eeg-based brain-computer interfaces," 2016.

[17] S. S. Arslan and P. Sinha, "Information transfer rate in bcis: Towards tightly integrated symbiosis," 2023.

[18] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from eeg with deep recurrent-convolutional neural networks," 2015.

[19] J. Chen, Y. Zhang, Y. Pan, P. Xu, and C. Guan, "A transformer-based deep neural network model for ssvep classification," Neural Networks, vol. 164, p. 521–534, July 2023.

[20] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "Eeg-gan: Generative adversarial networks for electroencephalograhic (eeg) brain signals," 2018.

Appendix

A MATLAB Code for CCA with PS input

```
% load_data.m, data loading function
function [eeg_data, freqs, phases] = load_data(dir,subjetcs)
    load(dir + "/Freq_Phase.mat");
    % load(dir + "/64-channel_locations.txt");
    % load(dir + "/subject_info_35_dataSets.txt");

    for subject = 1:size(subjetcs,2)
        eeg_data(subject) = load(dir + "/S" + subjetcs(subject) + ".mat
           ");
        disp(dir + "/S" + subjetcs(subject) + ".mat")
    end
    [eeg_data.eeg] = eeg_data.data; eeg_data = orderfields(eeg_data
       ,[1:0,2,1:1]); eeg_data = rmfield(eeg_data,'data');
end
```

```
% main.m, main file to be run
clear all;
close all;

data_dir = ~/SSVEP-BCI-Data"; % insert directory of SSVEP data
subjects = 1:35;
% subjects = [3];

[data, freqs, phases] = load_data(data_dir,subjects);

% Channel Selection
% channels = [61 62 63]; % O1, Oz, O2, according to '64-
   channel_locations.txt'
channels = [62];

for subject = 1:size(subjects,2)
    sel_ch_data(subject).eeg = data(subject).eeg(channels,:,:,:);
end

% Preprocessing
```

```
% bandpass filter of width 'wpass' in Hz
fs = 250;
wpass = [5, 100];

for subject = 1:size(subjects,2)
    for channel = 1:size(channels,2)
        for target = 1:size(sel_ch_data(subject).eeg, 3)
            for block = 1:size(sel_ch_data(subject).eeg, 4)
                disp("Subject: " + subject + " Channel: " + channel + "
                    Target: " + target + " Block: " + block)
                filtered_bandpass(subject).eeg(channel,:,target,block)
                   = bandpass(sel_ch_data(subject).eeg(channel,:,target
                   ,block),wpass,fs);
            end
        end
    end
end

% check PS, verify filtering
N = size(filtered_bandpass(1).eeg(1,:,8,1),2);
fs = 250;
ts = 1/fs;
t = ts*(0:N-1);
f = (0:N-1)*(fs/N);     % frequency range
f_trunc = f(1:200);
% %%
% figure;
% for plt = 1:6
%     if plt <= 3
%         y = sel_ch_data(1).eeg(plt,:,25,3);
%         Y = fft(y);
%         Y = abs(Y).^2/N;
%         Y_trunc = Y(1:200);
%
%         subplot(6,1,plt); plot(f_trunc,Y_trunc);
%     else
%         y = filtered_bandpass(1).eeg(plt-3,:,25,3);
```

```
%          Y = fft(y);
%          Y = abs(Y).^2/N;
%          Y_trunc = Y(1:200);
%
%          subplot(6,1,plt); plot(f_trunc,Y_trunc);
%      end
% end
%
% xlabel("Frequency response (Hz)");


%% CCA with PS as input
close all;

ch = 1; % Oz

for sub = 1:size(subjects, 2)
    disp("Subject: " + sub);
    PS_EEG(sub).Y = zeros(size(filtered_bandpass(sub).eeg, 1), size(
       filtered_bandpass(sub).eeg, 2), size(filtered_bandpass(sub).eeg,
        3), size(filtered_bandpass(sub).eeg, 4));

    % FT of input EEGs
    for freq = 1:size(freqs,2)
        for block = 1:size(filtered_bandpass(sub).eeg, 4)
            y = filtered_bandpass(sub).eeg(ch,:,freq,block);
            Y = fft(y);
            PS_EEG(sub).Y(ch,:,freq,block) = abs(Y).^2/N;
        end
    end

    % create reference signals (PSs)
    num_harm = 5;

    y_f_ccaps = zeros(num_harm,1500,size(freqs,2));
    for freq = 1:size(freqs,2)
        for harm = 1:num_harm
            y_f_ccaps(harm,:,freq) = cos(2*pi*freqs(freq)*(harm)*t);
```

```
        Y_f_ccaps(harm,:,freq) = fft(cos(2*pi*freqs(freq)*(harm)*t)
           );
        Y_f_ccaps(harm,:,freq) = abs(Y_f_ccaps(harm,:,freq)).^2/N;
    end
end

% CCA
r_t = zeros(size(PS_EEG(sub).Y, 3), size(PS_EEG(sub).Y, 3), size(
   PS_EEG(sub).Y, 4));
r_max = zeros(size(PS_EEG(sub).Y, 4), size(PS_EEG(sub).Y, 3));
for block = 1:size(PS_EEG(sub).Y, 4)
    for freq_eeg = 1:size(freqs,2)
        for freq_ref = 1:size(freqs,2)
            [A,B,r(freq_eeg,freq_ref),U,V] = canoncorr(PS_EEG(sub).
               Y(ch,:,freq_eeg,block)',Y_f_ccaps(:,:,freq_ref)');
        end
    end
    r_t(:,:,block) = r(:,:);
    [M,r_max(block,:)] = max(r');
end

% Mean rho
r_t_mean = mean(r_t,3);
[M,r_max_mean] = max(r_t_mean');

% reorder freqs for better visualization
for row = 1:6
    k = 1;
    for j = 0:7
        for i = 1:8:40
            r_new(row,k) = r_max(row,i+j);
            r_mean_new(k) = r_max_mean(i+j);
            freqs_new(k) = freqs(i+j);
%             disp("row " + row + " j " + j + " i " + i + " k " + k
    + " i+j " + (i+j) + "  element " + r_max(row,i+j))
            k = k + 1;
        end
    end
```

```
end

for i = 1:6
    for j = 1:40
        r_new_2(i,j) = find(freqs_new == (freqs(r_new(i,j))));
        r_mean_new_2(j) = find(freqs_new == (freqs(r_mean_new(j))))
            ;
    end
end

hit_count = 0;
for i = 1:size(freqs_new,2)
    if r_mean_new_2(i) == i
        hit_count = hit_count + 1;
    end
end

accuracy(sub) = hit_count/size(freqs_new,2);

figure;
plot(r_mean_new_2, 1:40, '.'); hold on;
plot(0:40, 0:40);
title("CCA with PS Mean rho value maxima | Subject: " + sub);
xlabel("Predicted frequency");
ylabel("Ground truth frequency");

for pred = 1:40
    for grd_trth = 1:40
        r_3D(grd_trth,pred) = size(r_new_2(r_new_2(:,grd_trth) ==
            pred),1);
    end
end

figure;
surf(r_3D)
colorbar
view(2)
title("CCA with PS Count of rho value maxima | Subject: " + sub);
```

```
    xlabel("Predicted frequency");
    ylabel("Ground truth frequency");
end

% ITR
for sub = 1:size(subjects,2)
    if accuracy(sub) == 1
        ITR(sub) = 60 / 5 * (log2(40));
    else
        ITR(sub) = 60 / 5 * (log2(40) + accuracy(sub)*log2(accuracy(sub
           )) + (1 - accuracy(sub))*log2((1 - accuracy(sub))/(40 - 1)))
           ;
    end
end
```

B Python Code

Python was the primary programming language utilized for implementing various preprocessing and classification methodologies. This included the development of scripts and algorithms for the CCA, FBCCA, and deep learning models.