UPPSALA
UNIVERSITET

IT 20 089
Examensarbete 30 hp
December 2020

Decoding Steady-State Visual Evoked Potentials(SSVEPs)

- Implementation and Performance Analysis

Peipei Han

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Decoding Steady-State Visual Evoked Potentials(SSVEPs)

Peipei Han

Steady-state visual evoked potential (SSVEP)-based brain-computer interfaces(BCIs) have been widely investigated. Algorithms from the canonical correlation analysis(CCA) family perform extremely well in detecting stimulus targets by analyzing the relationship of frequency features between electroencephalogram (EEG) signals and stimulus targets. In addition to CCA algorithms, convolutional neural networks(CCNs) also improve the performance of SSVEP-based BCIs by generalizing well on the frequency features of the EEG signals. To find a new method for speeding up an online SSVEP decoding system, we have evaluated three CCA methods which are standard CCA, individual-template CCA(IT-CCA), and Extended CCA, together with the complex spectrum CNN(C-CNN). The results have proved that algorithms requiring individual subject training highly outperform standard CCA.

Handledare: Mohammad Davari
Ämnesgranskare: Carolina Wählby
Examinator: Mats Daniels
IT 20 089
Tryckt av: Reprocentralen ITC

Contents

1 Introduction

In recent years, brain signal processing has become a popular area of research in the field of signal processing. Along with the development of electroencephalogram (EEG) recording devices, Brain-computer interfaces (BCIs) processing EEG signals have received widespread attention in neural engineering, neuroscience, and clinical rehabilitation communities. Steady-state visual evoked potential (SSVEP)-based BCI as one of the earliest BCIs is still one of the most promising types of BCIs because they are non-invasive, easy to use, and relatively low-cost. In a typical SSVEP-based BCI system, several targets flash at a different frequency where each one associates with a command controlling outside devices. When the user gazes at one of the targets, various algorithms can determine the target which the subject is focusing on. Thus, SSVEP-based BCIs can be used as a special communication system to help people with severe disabilities such as stroke, paralyze, and amyotrophic lateral sclerosis(ALS). To boost the performance of SSVEP-based BCIs, researchers are working on two main directions: (1) improving the design of stimulus, (2) improving EEG decoding algorithms. In this study, we mainly focus on evaluating four target classification algorithms on a public dataset with 12 subjects. The algorithms we evaluated are standard canonical correlation analysis(CCA), Individual Template CCA(IT-CCA), Extended CCA which combines the standard CCA and IT-CCA. Since convolutional neural networks(CNNs) have attracted much attention and scored great achievements in the visual field, we have also designed a CNN model to analyze the complex spectrum features of the EEG signals. The source code can be found at https://github.com/peipeihan2020/SSVEP.git .

2 Background

2.1 Brain-Computer Interface(BCI)

A Brain-Computer Interface(BCI) is a system that provides a communication pathway between the brain and electrical devices.[8]. BCIs can help people with severe disabilities such as stroke, amyotrophic lateral sclerosis(ALS), or paralysis to act by brain signals rather than damaged muscles. A BCI transforms brain signals into commands that can be recognized by the electrical devices. For example, a paralyzed person can control a motorized wheel-chair simply by brain activities through a BCI system. Figure 1 presents five types of BCI applications [8] :

- The output of a BCI system can replace the lost output of the brain due to injury or decease. For example, a person who lost speaking ability can use a BCI system to spell the words on a computer.

- A BCI output can restore the lost output of the brain. For example, a person with spinal cord injury which causes paralyzed arms and hands can implant a BCI system to stimulate the paralyzed limbs to make them work again.
- A BCI output might enhance the output of normal brain signals. For example, a BCI system might be able to make people more concentrated during study or work.
- A BCI output might supplement the output of brain activities. For example, a normal person might use a BCI system to control a third robotic arm.
- A BCI output might improve the output of brain signals. For example, a person with impaired arm movements due to stroke can use a BCI system to control an extra device which helps to move the arms.

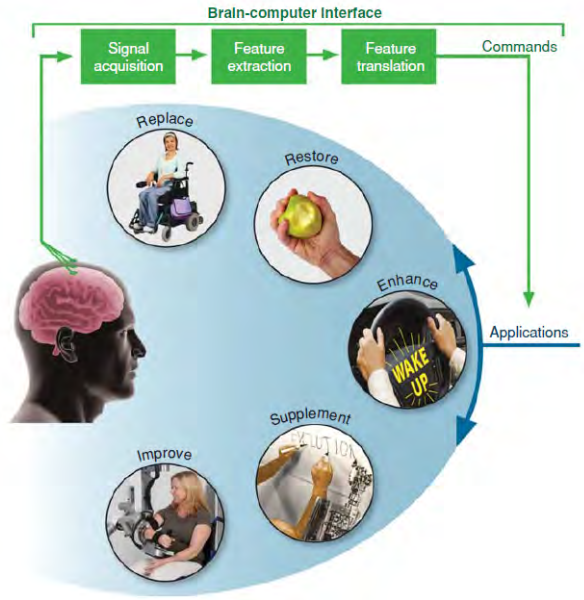


Figure 1: The basic design and operation of a brain-computer interface (BCI) system. In this illustration, the BCI is shown in green. Electrical signals reflecting brain activity are acquired from the scalp, from the cortical surface, or from within the brain. The signals are analyzed to measure signal features (such as amplitudes of EEG rhythms or firing rates of single neurons) that reflect the user's intent. These features are translated into commands that operate application devices that replace, restore, enhance, supplement, or improve natural brain outputs[8]. (Modified from Wolpaw et al., 2002.) (Supplement image © Stelarc, http://stelarc.org ; Improve image © Hocoma AG, www.hocoma.com .)

As shown in Figure 1, a BCI system usually consists of signal acquisition as shown in Section 2.1.1, feature extraction, feature translation, and applications as we illustrated in Section 2.1.

2.1.1 Signal Acquisition

To transform the acquired brain signals to commands which can be used in outside devices, the devices or technologies that are used to obtain brain signals must meet some requirements:

- Must be safe.
- Must contain sufficient information for later decoding.
- Must be reliable.

Brain activity involves three types of processes: electrical, chemical, and metabolic, all of which can be measured and recorded by modern brain imaging technologies. Action potentials and other neuronal electrical phenomena allow for electrical measurement of brain activity with electroencaphalography (EEG), magnetoencephalography (MEG), electrocorticography (ECoG), or with microelectrodes implanted in the brain tissue. The chemical process of brain activity can be measured by positron emission tomography(PET) which is still rather crude, involving the injection of specially manufactured markers and have the poor temporal resolution. Metabolic processes involve the utilization of energy which increases the demand for glucose and oxygen. This change is accompanied by increased blood flow to the region. The blood flow can be measured by functional magnetic resonance imaging(fMRI) which gives good spatial resolution but moderate temporal details[8]. Among all these brain imaging technologies, noninvasive EEGs are widely used in various BCI systems because of their high portability, high temporal resolution, low cost, easy to use, and most importantly their non-invasiveness.
An EEG recording system consists of electrodes, amplifiers, analog-to-digital converters, and a signal acquisition device. The EEG system measures the potential difference between signal electrodes and a reference electrode and voltage difference between signal electrodes and the reference electrode using ground electrodes. These electrodes are commonly placed at locations on the scalp based on the International 10-20 system as shown in Figure 2, where "10" and "20" refer to the distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left distance of the skull [4].

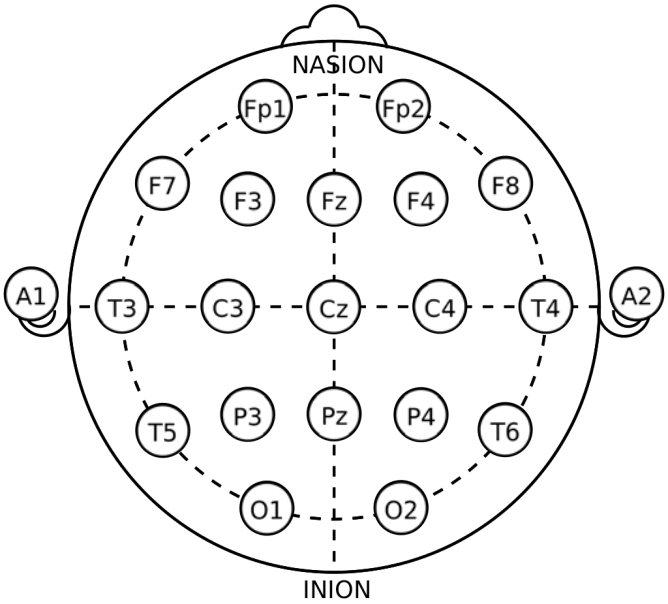


Figure 2: The International 10-20 system for EEG [9]

2.2 Steady-state Visual Evoked Potential(SSVEP)

When a subject is presented with a visual stimulus, light information is recorded by the photoreceptors of the retina and passed to the retinal ganglion cells which are neurons that fire action potentials, in response to the visual stimulation. These action potentials are propagated through the optic nerves to the visual cortex and other regions of the brain, thereby can be captured at the occipital area of the scalp using EEG which consists of the positive and negative deflections of the visual-evoked potentials(VEPs)[10]. When the visual stimulus flicks at a repetitive frequency of at least 4 Hz, these deflections generate a steady-state visual-evoked potential(SSVEP) with frequency and phase locked to the stimulus[11]. Especially when the subject focuses on the stimulus for a short period time, enhanced responce can be observed in the SSVEP, a study [12] has proved that voluntarily attention directed to a spatial location enhances the evoked potentials. By measuring the frequency and phase of the EEG signals, we can determine which stimulus the subject is attending. This is the basis of SSVEP-based (BCI).

2.3 Filter

A filter in signal processing is a process that keeps wanted components and removes undesired components from the signals. Filters can be classified into four types based on their tasks in the frequency domain: high pass, low pass, band pass

and band stop as shown in Figure 3. High pass filters remove or attenuate low-frequency signals, low pass filters remove or attenuate high-frequency signals, band pass filters only keep signals with frequency in the pass range and band stop filters remove or attenuate signals with frequencies in the stop range [1].

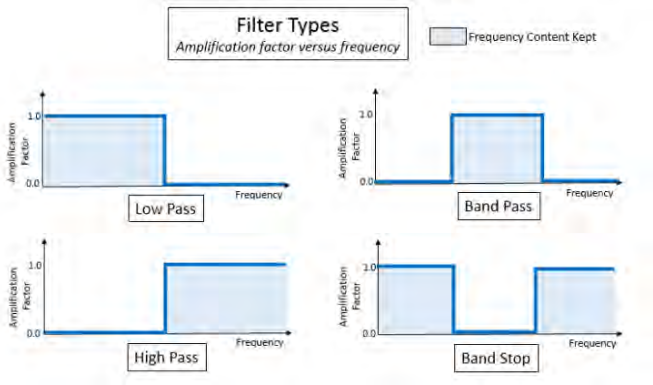


Figure 3: Four Types of Filters [1]

Every digital signal comprises of a large number of sine and cosine waves with different frequencies. A perfect filter keeps the pass-band frequencies of the signal unchanged and removes the stop-band frequencies entirely from the input to output. This means a flat frequency response in the pass-band, a zero roll-off response in the stop-band, and an infinitely sharp roll-off in the transition band[13].
However, such perfect filters do not exist. There are two categories of filters which can approximate the ideal behavior, each with advantages and disadvantages: the Finite Impulse Response(FIR) filters and the Infinite Impulse Response(IIR) filters, where the impulse response indicates the output waveform corresponding to the input signal presenting in the time domain.

2.3.1 Finite Impulse Response(FIR) filter [1]

The FIR filter is non-recursive filter: it calculates the output sample by accumulating the current input sample of the signal together with previous input samples.

$$y(n) = \sum_{k=0}^{N} a(k)x(n-k) \tag{1}$$

where n is the number of data point in the input signals. x(n) is the input signal. N is the order of the filter(the maximum number of delayed elements used to calculate the current output signal). y(n) is the output signal. a(k) is the weighted coefficients. Because N is a limited number, the output can be a product of

the input signal and a window of length N. The higher the order(N), the more expensive the computation and the sharper the roll-off in the transition band will be. Furthermore, the type of window influences the amplitudes in pass-band and stop-band. The amplitude will present a ripple. In the pass-band, the amplitude will be less than 1 but slightly different for every frequency. The larger the ripple, the larger the amplitude differences will be, and the more the signal will degrade at the output. In the stop-band, most undesired frequencies will in some amount still be present in the output signal. The length of the window will also determine the delay. When a signal is fed into an FIR filter, the output will have shifted its phase compared to the input. The phase shift is equal to half the length of the window, multiplied by the sample time; and each frequency in the signal will be shifted with the same constant phase shift.

FIR filter advantages

- Its output is stable because it is non-recursive.
- Its delay is constant.

FIR filter disadvantages

- It uses lots of input to calculate the output, therefore inefficient and has a large delay.

Zero Phase Filtering The delay in the output time history can be eliminated by filtering the data "forward and backward". After the time history x(n) is filtered, and the new output time history y(n) is created, it can be fed back into the filter. The data points in y(n) are reversed in order in time to do this and fed into the filter again. However, the calculation takes twice as long to perform and data is eliminated at end of time trace.

In the context of EEG signal processing, FIR filters can be used for offline applications as high pass or low pass filters with a high order to attain a sharp roll-off in the transition band since a large delay is allowed. These filters are especially well suited if the phase characteristic of the signal is very important and can thus be used for coherence and synchronous measures.

2.3.2 Infinite Impulse Response(IIR) filter [1]

An IIR filter is recursive which depends linearly on a finite number of input samples and a finite number of previous filter outputs. In other words, it combines an FIR filter with feedback from previous filter outputs. Mathematically, for some coefficients a(k) and b(j):

$$y(n) = \sum_{k=0}^{N} a(k)x(n-k) + \sum_{j=0}^{P} b(j)y(n-j) \tag{2}$$

Where N is the order of the filter. As was the case with FIR filters, increasing the order of the filter will improve the amplitude characteristics of the filter. Naturally, there is a drawback to increasing filter order. More samples are used to calculate output, therefore the filter is less efficient. This results in degradation of the phase characteristic.

IIR filter advantages

- An IIR filter is very efficient, it uses only a few previous output values to calculate the current output, therefore: An IIR filter has only a small delay.

IIR filter disadvantages

- An IIR filter can become unstable if coefficients are not well chosen.
- Because of the requirement of causality and an infinite impulse response that converges to 0 for $n => \infty$, the impulse response cannot be symmetrical. Therefore, the delay in output signals varies.

The use of an IIR filter is appropriate when a small absolute delay is required or when strong requirements on amplitude characteristics are necessary. There is only one type of FIR filter, while there are several types of IIR filters. In this study, we have investigated three of them: the Butterworth, the Chebyshev, and the Elliptic IIR filters.

- Butterworth IIR[14]
  A Butterworth filter has a maximally flat pass-band [13] and rolls off towards zero in the stop-band at the expense of a wide transition band from the passband to the stopband and a poor phase characteristics. So Butterworth filter is an appropriate choice when a maximally flat pass- and stop-bands matter most and fall-off or phase characteristics contribute no or little to the result. For example, Butterworth filters can be used in an EEG frequency band analysis. The pass- and stop-bands are maximally flat, thus resulting in a quality output signal for the different frequency bands. Furthermore, a reasonably sharp fall-off can be reached by choosing an appropriate filter order. On the other side, with poor phase characteristics, synchronization, or other phase-related analysis is not recommended with this filter.
- Chebyshev IIR [14]
  A Chebyshev filter has a constant ripple in the pass-band. The number of ripples can be adjusted by filter order. The bigger the filter order, the more ripple will be. In comparison with a Butterworth filter, the stop-band behavior and fall-off of a Chebyshev filter are better. The same as the Butterworth filter, the phase characteristic of a Chebyshev filter is also poor.

Therefor a Chebyshev IIR filter is appropriate when certain ripples in the passband are allowed and more emphasis is placed on good stop-band behavior and fall-off.

- Elliptic IIR [14]
  An elliptic IIR filter has constant ripples in both pass-band and stop-band which increase with filter order. In comparison with the Butterworth as well as the Chebyshev, at a given order, the fall-off is better for an elliptic filter. Therefore an elliptic IIR filter is most appropriate when the fall-off is very critical.

2.4 Canonical Correlation Analysis(CCA)

Canonical correlation analysis(CCA) is a method for measuring the relationship between two sets of multivariate variables[15]. It has been widely used to detect the target frequency of SSVEPs because of its easy-to-use and computational efficiency. For two sets of variables, we can also calculate the pairwise correlation from each set to determine their relationship. However, by doing so, it is relatively computational expensive and very difficult to interpret the result. CCA instead calculates the correlation of the linear combination of each set which is more effective and able to preserve most facets of the relationship[3].
Considering two sets of variables X and Y which have p and q variables, respectively:

$$X = \left(X_1, X_2, ..., X_p\right)^T$$

$$Y = \left(Y_1, Y_2, ..., Y_q\right)^T$$

and for convenience, suppose $p <= q$. CCA comprises the following steps:

2.4.1 The variance of Linear Combination of Multivariate[2]

Let the linear combination of X be U with coefficients $a_1, a_2, ...a_p$ corresponding to the p X variables, where

$$\begin{aligned} U &= a_1X_1 + a_2X_2 + ...a_pX_p \\ &= \sum_{i=1}^{p} a_iX_i \end{aligned}$$

The variance of U σ_u^2 can be calculated by:

$$\begin{aligned}
\sigma_u^2 &= E((\sum_{i=1}^{p} a_i X_i - \overline{U})^2) \\
&= E((\sum_{i=1}^{p} a_i X_i - E(\sum_{i=1}^{p} a_i X_i))^2) \\
&= E((\sum_{i=1}^{p} a_i X_i)^2 - 2\sum_{i=1}^{p} a_i X_i) E(\sum_{i=1}^{p} a_i X_i) + E(\sum_{i=1}^{p} a_i X_i)^2) \\
&= E((\sum_{i=1}^{p} a_i X_i)^2) - E(\sum_{i=1}^{p} a_i X_i)^2 \\
&= E(\sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j X_i X_j) - E(\sum_{i=1}^{p} a_i X_i)^2 \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j E(X_i X_j) - E(\sum_{i=1}^{p} a_i X_i)^2 \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j E(X_i X_j) - E(\sum_{i=1}^{p} a_i X_i) E(\sum_{i=1}^{p} a_i X_i) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j E(X_i X_j) - \sum_{i=1}^{p} a_i E(X_i) \sum_{i=1}^{p} a_i E(X_i) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j E(X_i X_j) - \sum_{i=1}^{p}\sum_{j=1}^{n} a_i a_j E(X_i) E(X_j) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j (E(X_i X_j) - E(X_i) E(X_j)) \\
&= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j cov(X_i, X_j),
\end{aligned}$$

where $\overline{U} = \sum_{i=1}^{n} a_i \hat{X}_i$ indicates the mean of U, and $cov(X_i, X_j) = E((X_i - E(X))(X_j - E(X)))$ denotes the covariance between X_i and X_j .

Similarly, let the linear combination of Y be V with coefficients $b_1, b_2, ...b_q$ corresponding to the q Y variables, where

$$V = b_1Y_1 + b_2Y_2 + ...b_qY_p$$
$$= \sum_{i=1}^{q} b_iY_i$$

The variance of V σ_v^2 is:

$$\sigma_v^2 = \sum_{i=1}^{q}\sum_{j=1}^{q} b_ib_jcov(Y_i, Y_j),$$

where $cov(Y_i, Y_j) = E((Y_i - E(Y))(Y_j - E(Y)))$ denotes the covariance between Y_i and Y_j .

2.4.2 Covariance of Linear Combination of Multivariate

The covariance of U and V $cov(U, V)$ is:

$$cov(U, V) = E((U - \overline{U})(V - \overline{V}))$$
$$= E((\sum_{i=1}^{p} a_iX_i - \sum_{i=1}^{p} a_iE(X_i))(\sum_{j=1}^{q} b_jY_j - \sum_{j=1}^{q} b_jE(Y_j)))$$
$$= E(\sum_{i=1}^{p}\sum_{j=1}^{q} a_ib_jX_iY_j - \sum_{i=1}^{p}\sum_{j=1}^{q} a_ib_jX_iE(Y_j)) - \sum_{j=1}^{q}\sum_{i=1}^{p} a_ib_jE(X_i)Y_j + \sum_{i=1}^{p}\sum_{j=1}^{q} a_ib_jE(X_i)E(Y_j)))$$
$$= \sum_{i=1}^{p}\sum_{j=1}^{q} E(X_iY_j) - \sum_{i=1}^{p}\sum_{j=1}^{q}(E(X_i)E(Y_j))$$
$$= \sum_{i=1}^{p}\sum_{j=1}^{q}(E(X_iY_j) - E(X_i)E(Y_j))$$
$$= \sum_{i=1}^{p}\sum_{j=1}^{q} a_ib_jcov(X_i, Y_j)$$

where $\overline{U}, \overline{V}$ are the mean of U,V respectively, and $cov(X_i, Y_j)$ is the covariance between X_i and Y_j.

2.4.3 Correlation of Linear Combination of Multivariate

The correlation between U and V $\rho(U, V)$ is:

$$\begin{aligned}\rho(U,V) &= \frac{cov(U,V)}{\sqrt{\sigma_u^2}\sqrt{\sigma_v^2}} \\ &= \frac{\sum_{i=1}^{p}\sum_{j=1}^{q} a_i b_j cov(X_i, Y_j)}{\sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j cov(X_i, X_j)}\sqrt{\sum_{i=1}^{q}\sum_{j=1}^{q} b_i b_j cov(Y_i, Y_j)}}.\end{aligned}$$

2.4.4 Maximum Correlation

A bigger correlation indicates a higher similarity between two sets of multivariates. CCA finds the weight vectors a and b which maximize the correlation between U and V by solving the following problem:

$$\rho(U,V)^* = max(\frac{\sum_{i=1}^{p}\sum_{j=1}^{q} a_i b_j cov(X_i, Y_j)}{\sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j cov(X_i, X_j)}\sqrt{\sum_{i=1}^{q}\sum_{j=1}^{q} b_i b_j cov(Y_i, Y_j)}}),$$

which can be simplified as:

$$\rho(U,V)^* = max(\frac{a^T \sum_{XY} b}{\sqrt{a^T \sum_{XX} a}\sqrt{b^T \sum_{YY} b}}).$$

To avoid the simultaneous explosion of a and b from both divisor and dividend, we add two constraints $a^T \sum_{XX} a = 1$ and $b^T \sum_{YY} b = 1$. Therefore, it becomes a constraint optimization problem and can be solved by Lagrangian:

$$\zeta = a^T \sum_{XY} b - \frac{\lambda}{2}(a^T \sum_{XX} a - 1) - \frac{\theta}{2}(b^T \sum_{YY} b - 1)$$

Differentiate it and get:

$$\begin{aligned}\frac{\partial \zeta}{\partial a} &= \sum_{XY} b - \lambda \sum_{XX} a = 0 \\ \frac{\partial \zeta}{\partial b} &= \sum_{XY} a - \theta \sum_{YY} b = 0\end{aligned} \tag{3}$$

Multiplying $\frac{\partial \zeta}{\partial a}$ with a^T and $\frac{\partial \zeta}{\partial b}$ with b^T gets:

$$\lambda = \theta = a^T \sum_{XY} b,$$

which is the covariance between X and Y. So this problem becomes to find the maximum λ. We simplify equation 3 as following on the assumption that the variance of X and Y is invertible:

$$\begin{aligned}(\sum_{XX})^{-1}\sum_{XY} b = \lambda a \\ (\sum_{YY})^{-1}\sum_{XY} a = \lambda b\end{aligned} \tag{4}$$

Then transform it to a matrix:

$$\begin{bmatrix}(\sum_{XX})^{-1} & 0 \\ 0 & (\sum_{YY})^{-1}\end{bmatrix}\begin{bmatrix}(0 & \sum_{XY} \\ \sum_{XY} & 0\end{bmatrix}\begin{bmatrix}a \\ b\end{bmatrix} = \lambda\begin{bmatrix}a \\ b\end{bmatrix} \tag{5}$$

Let $A = \begin{bmatrix}(\sum_{XX})^{-1} & 0 \\ 0 & (\sum_{YY})^{-1}\end{bmatrix}$, $B = \begin{bmatrix}(0 & \sum_{XY} \\ \sum_{XY} & 0\end{bmatrix}$ and $w = \begin{bmatrix}a \\ b\end{bmatrix}$.
Then convert equation 5 to:

$$A^{-1}Bw = \lambda w.$$

At this step, it becomes a finding eigenvalue and eigenvector problem.
The canonical correlation is a specific type of correlation. The maximum of $\rho(U, V)$ λ concerning a and b is the canonical correlation. Projections onto a and b are called canonical variants. In SSVEPs, X refers to the set of multi-channel EEG signals and Y refers to the set of reference signals that have the same length as X which can be denoted by $Y_n \in R^{2N_h \times N_s}$:

$$Y_n = \begin{bmatrix} sin(2_n t) \\ \\ cos(2_n t) \\ \\ \cdot \\ \cdot \\ \cdot \\ sin(2_h f_n t) \\ \\ cos(2_h f_n t)\end{bmatrix}, t = \left[\tfrac{1}{f_s}, \tfrac{2}{f_s} ..., \tfrac{N_s}{f_s}\right], \tag{6}$$

where n is the number of reference signals, N_h is the number of the harmonics, N_s is the number of samples, f_n is the reference signal and f_s is the sampling frequency. To recognize the stimulation frequency of the SSVEPs, CCA calculates the canonical correlation ρ_n between the multi-channel EEG signals X and the reference signals at each stimulus frequency Y_n. The frequency of the reference signals with the maximum correlation is selected as the target stimuli of the SSVEPs.

2.5 Individual Template based CCA(IT-CCA)[3]

Unlike standard CCA which finds the maximum canonical correlation between the EEG signals and the reference targets. IT-CCA recognizes the target stimuli of the SSVEPs by finding the maximum correlation between the multi-channel EEG data X and the test template $\overline{X}$ gathered during training time. For each target i, the template $\overline{X}_i \in R^{N_c \times N_t}$ can be obtained by averaging N training trials as $\overline{X}_i = \frac{1}{N}\sum_{h=1}^{N} X_h$, where N_c indicates the number of channels, N_t indicates the number of EEG samples within the time window t and X_h is the EEG signals at trail h. Then the reference frequency of the SSVEPs can be obtained by finding the maximum canonical correlation between current EEG signals with the template signals by calculating:

$$\rho_n = max(\frac{a^T \sum_{X\overline{X}} b}{\sqrt{a^T \sum_{XX} a}\sqrt{b^T \sum_{\overline{X}\overline{X}} b}}) \tag{7}$$

2.6 Extended CCA-based method[4]

The extended CCA-based method combines IT-CCA and standard CCA approaches to generate a canonical correlation vector for each target frequency. It obtains three weight vectors from IT-CCA and standard CCA:

- $W_x(X, \overline{X}_n)$ is the weight vector for test EEG signals obtained by calculating the canonical correlation between the test EEG signals X and the individual template $\overline{X}_n$, where n is the number of the target frequency.
- $W_x(X, Y_f)$ is the weight vector for test EEG signals obtained by calculating the canonical correlation between the test EEG signals X and the stimulate signals Y_f.
- $W_{\overline{X}_n}(\overline{X}^{(n)}, Y_f)$ is the weighted vector for the individual template obtained by computing the canonical correlation between individual template $\overline{X}_n$ and the stimulate signals Y_f.

A correlation vector r_n is defined as follows:

$$r_n = \begin{bmatrix} r_{n,1} \\ r_{n,2} \\ r_{n,3} \\ r_{n,4} \\ r_{n,5} \end{bmatrix} = \begin{bmatrix} \rho(X, Y_n) \\ \rho(X, \overline{X}) \\ \rho(X^T W_x(X, Y_f), \overline{X}_n^T W_x(X, \overline{X}_n)) \\ \rho(X^T W_{\overline{X}_n}(\overline{X}_n, Y_f), \overline{X}_n^T W_{X_n}(\overline{X}_n, Y_f)) \\ \rho(\overline{X}_n^T W_x(X, \overline{X}_n), \overline{X_n} W_{\overline{X}_n}(X, \overline{X}_n)) \end{bmatrix} \tag{8}$$

where $\rho(a, b)$ indicates the Pearson's correlation coefficient between a and b. An ensemble classifier $\hat{r}_n$ is used to combine the five features and get the final target.

$$\hat{r}_n = \sum_{i=1}^{5} sign(r_{n,i})(r_{n,i}^2)$$

where $sign()$ is used to retain discriminative information from negative correlation coefficients. The target can be identified with maximum $\hat{r}$.

2.7 Complex Spectrum Feature

2.7.1 Fourier Series[5]

A Fourier series is the representation of a periodic function as a (possibly infinite) sum of sinusoidal functions.
The Fourier series of a periodic function f(x) is:

$$f(x) = \sum_{n=0}^{\infty} a_n cos(\frac{2\pi n}{L} x) + \sum_{n=0}^{\infty} b_n sin(\frac{2\pi n}{L} x), \tag{9}$$

where L is the period of the signal with the coefficients a_n and b_n defined by the inner products:

$$\begin{aligned} a_0 &= \frac{1}{2L} \int_0^L f(x) dx, \\ a_n &= \frac{2}{L} \int_0^L f(x) cos(\frac{2\pi n}{L} x) dx, n > 0, \\ b_n &= \frac{2}{L} \int_0^L f(x) sin(\frac{2\pi n}{L} x) dx. \end{aligned}$$

Since Fourier series decompose functions into sine and cosine functions, we can use Euler's formula $e^{inx} = cos(nx) + isin(nx)$ to write a Fourier series in the complex form with complex coefficients $c_n = \alpha_n + i\beta_n$:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx} \tag{10}$$

2.7.2 Fourier Transform[6]

The Fourier series is defined for periodic functions so that outside the domain of definition, the function repeats itself forever. The Fourier transform integral is essentially the limit of a Fourier series as the length of the domain goes to infinity, which allows us to define a function defined on $(-\infty,\infty)$ without repeating. the Fourier Transform transfers a time-based signal into a signal in the frequency domain. It decomposes any function into a combination of sinusoidal basis functions as shown in figure 4.

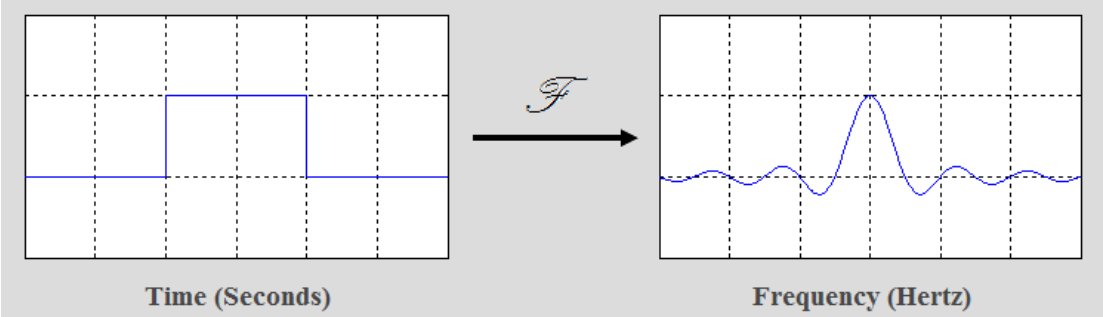


Figure 4: Fourier Transform

The Fourier Transform of a function g(t) is defined by:

$$G(f) = \int_{-\infty}^{\infty} g(t)e^{-2\pi ift}dt \tag{11}$$

where $G(f)$ denotes the spectrum of g and f indicates the frequency.

2.7.3 Discrete Fourier Transform(DFT)[6]

The Fourier series and transform are related to continuous function $f(x)$. However, when working with real data, it is necessary to approximate the Fourier Transform on discrete vectors of data. The resulting discrete Fourier transform (DFT) is essentially a discretized version of the Fourier series for vectors of data $f = [f1, f2, f3, ..., fn]^T$ obtained by discretizing the function f (x) at a regular

spacing δx.
The DFT is one of the most powerful tools in digital signal processing which enables us to find the spectrum of a finite-duration signal. The discrete Fourier transform (DFT) is given by:

$$\hat{f}_k = \sum_{j=1}^{n} f_j e^{-i2\pi jk/n} \tag{12}$$

Thus the DFT is a linear operator that maps the data points in f to the frequency domain $\hat{f}$:

$$f_1, f_2, ..., f_n \Longrightarrow \hat{f}_1, \hat{f}_2, ...\hat{f}_n.$$

2.7.4 Fast Fourier Transform(FFT)[6]

The time complexity of DFT is $O(n^2)$. For large n, the DFT is computationally expensive and thus inefficient. Fast Fourier transform (FFT) utilizes fractal symmetry in the Fourier transform that allows an n-dimensional DFT to be solved with serveral smaller dimensional DFT computations. Its time complexity is $O(nlog(n))$ which is much more efficient when n gets bigger.

2.7.5 Complex Spectrum Feature

The complex FFT representation was used to derive both magnitude and phase related information of a signal. First, the input time-domain signal was transformed into the complex FFT representation using the standard FFT. Next, the frequency components of the real part and the imaginary part along each channel were extracted between band-pass frequencies resulting in two vectors of the same length. These two vectors were concatenated into a single feature vector as:

$$I = R(X)||I(X)$$

where $R(X)$ contains the real part and $I(X)$ contains the imaginary part of the complex FFT resulting in a vector. The resultant feature vector was stacked together to form a matrix[4].

2.8 Complex Feature-Convolutional Neural Network(C-CNN)

Convolutional neural networks (CNNs) have gained high success in computer vision especially in image processing, where they can beat humans in some special tasks. The success has inspired studies of applying CNNs to classify EEG signals. For example, the study [16] uses frequency features as the input of two CNN classifiers for SSVEP classification. In this study, we use the Complex Spectrum feature of EEG signals as the input to a CNN model which is called the Complex Feature-CNN(C-CNN)[4].

2.8.1 Convolution Layer [7]

The convolution layer is the central block of CNNs. This layer performs a convolution operation which is a linear operation that involves the element-wise multiplication of a set of weights with the input where the set of weights is called a filter. The filter is usually designed to detect a specific type of feature in the input, then the application of that filter systematically across the entire input image allows the filter an opportunity to discover that feature anywhere in the image.

2.8.2 Dropout[7]

Dropout is one of the most popular regularization techniques and it is proven to be highly successful. The algorithm is very simple. During training, at every step, every neuron excluding the output neurons has a probability p(dropout rate) of being ignored. During the test, neurons don't get dropped anymore.

2.8.3 Batch Normalization(BN)[7]

Batch Normalization(BN) is a technique to address the vanishing/exploding gradients problems. BN is usually applied before or after the activation function of each hidden layer which consists of zero-centering and normalizing each input, then scaling and shifting the result using two new parameter vectors per layer: one for scaling, the other for shifting. In other words, this operation lets the model learn the optimal scale and mean of each of the layer's inputs[7].

2.8.4 K-Fold Cross Validation[7]

K-Fold cross validation is a very commonly used method in machine learning with a small size of training data where K denotes the number of groups to split the training data. To apply this method, first, divide the training data into K groups with equal size. For each group i, train a model on the remaining $K-1$ groups and evaluate it on the group i. Each sample in the training data is given an opportunity to be used in the hold- out set 1 time and used to train the model $K-1$ times. An example of a three-fold cross fold is shown in Figure 5.

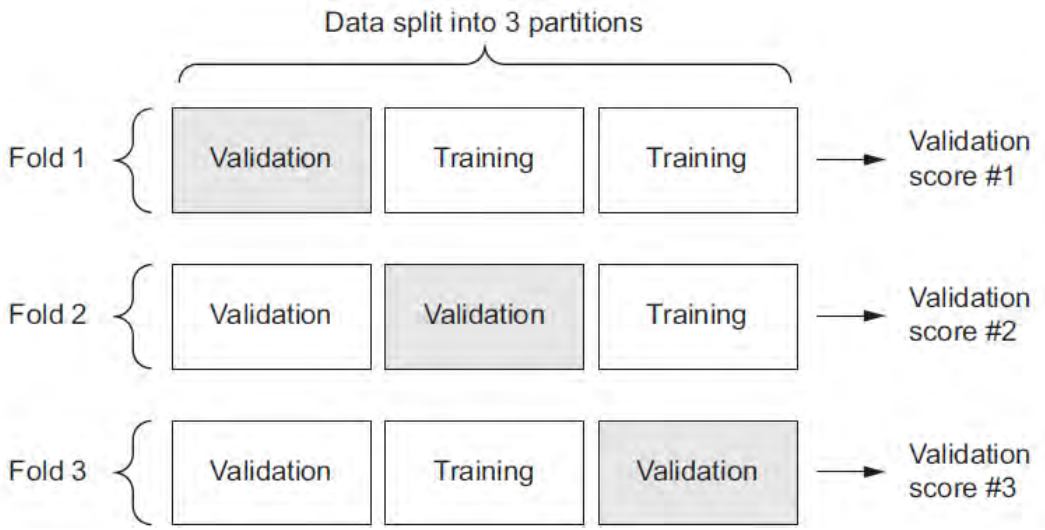


Figure 5: Threefold cross validation[17]

3 Experiments

3.1 Offline Experiment

3.1.1 Dataset

A public 12-class EEG dataset is used for offline experiments[3]. This dataset was recorded in an online SSVEP experiment with 10 healthy subjects including 9 males and 1 female whom all have a normal or corrected-to-normal vision. All subjects were seated in a comfortable chair $60cm$ in front of a $27inch$ ASUS VG278 LCD monitor with a refresh rate of 60Hz and a resolution of 1280×800 pixels in a dim room. In the monitor, twelve flicking rectangles arranged in a 4×3 matrix($5cm$ gap between each column and $1.5cm$ between each row) were presented to the subjects. The frequencies of all the stimulus were started from 9.25Hz till 14.75Hz with a 0.5Hz step increasing for each rectangle. Each flicker has the same phase in the same row while 0.5π bigger than the previous row as shown in Figure 6. The EEG signals were collected by eight Ag/AgCl electrodes attaching to the occipital area of each subject and then amplified and digitized by a BioSemi ActiveTwo EEG system (Biosemi, Inc.) with a sampling rate of 2,048Hz. All electrodes were regarding the CMS electrode close to Cz. Each subject's experiment consisted of 15 blocks. In each block, the subjects were asked to finish 12 trials for each target. At the beginning of each trial, one of the 12 flickers became red as shown in Figure 6 at random order for 1s. Subjects were required to switch gaze to the red target.

An onset event trigger was sent from the parallel port of the computer to the EEG system and recorded on an event channel synchronized to the EEG data. After 1s, all the stimulation started to flick simultaneously for 4s. During this period, the subjects are required to not blink their eyes to avoid introducing artifacts to the EEG signals.
After the experiments, data epochs comprising eight-channel were extracted according to the onset event triggers. All data epochs were down-sampled to 256Hz. Considering a latency delay in the visual system, the data epochs were extracted in [0.135s 0.135+d s], where the time 0 indicated stimulus onset and d indicated data length used in the offline analysis. The 135-ms delay was selected towards the highest classification accuracy.

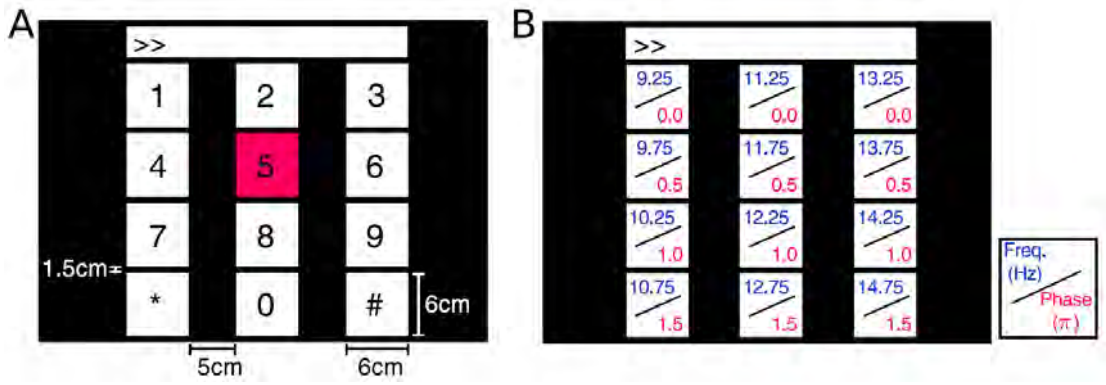


Figure 6: Stimulus design of the 12-target BCI system. (A) The user interface of a virtual keypad for a phone-dialing program. (B) Frequency and phase values are specified for each target. The red square in (A) is the visual cue indicating a target symbol '5' in the experiment.

3.1.2 Setting up of Filters

The filters we have experimented within this project are Butterworth, Chebyshev, and Elliptic which all have the same number of order 4 and pass-band between 6Hz and 80Hz. The maximum ripple allowed below unity gain in the pass-band for Chebyshev and Elliptic filters is both set to 0.3. While the extra parameter which is the minimum attenuation required in the stop-band of the Elliptic filter is set to 1.

3.1.3 Setting up of IT-CCA and Extended CCA

The number of training trails to get individual templates for IT-CCA and Extend CCA method are set to 14.

3.1.4 Architecture of the CNN Model

The CNN model consists of four layers, an input layer, two convolution layers, and a fully connected layer as shown in Figure 7 [4]. The input is the complex spectrum calculated by FFT from EEG signals with shape$(N_{ch}, N_{fc}, 1)$, where N_{ch} denotes the number of channels and N_{fc} denotes the number of frequency data samples. The input is passed to the first convolution layer $Conv_1$ and convoluted with a filter whose size is $(N_{ch}, 1)$. Then the output of layer $Conv\_1$ is passed to the second convolution layer $Conv\_2$ after being batch normalized and regularized by a dropout rate of 0.25. The filter size used in layer $Conv\_2$ is (1,10). The output of $Conv_2$ is handled by the same process as the output of $Conv\_1$ and then passed to the output layer.
We train the model using the SGD optimizer, minimizing the categorical cross-entropy loss function. Since the dataset is small, we apply 4-fold cross-validation to the data and report the final result as the average of results from all the four models.

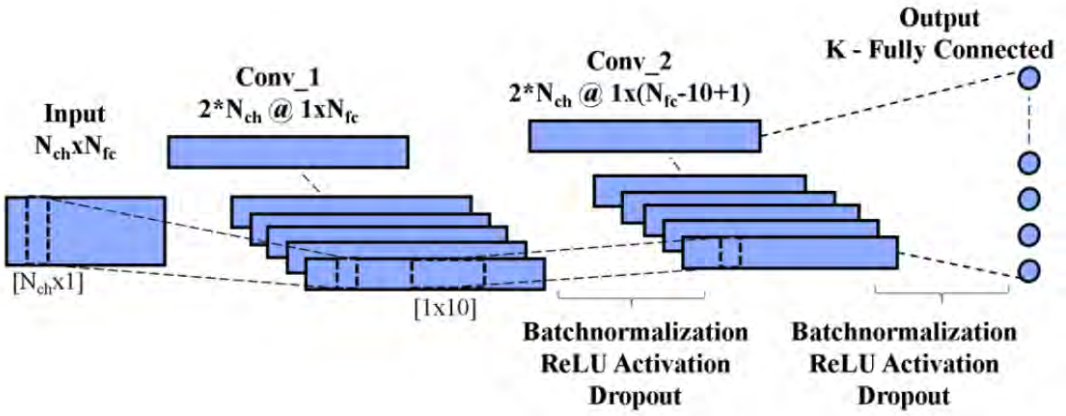
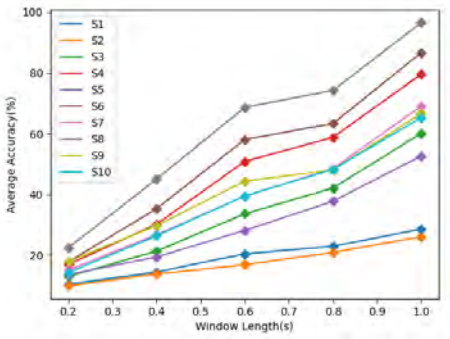


Figure 7: Architecture of CNN model

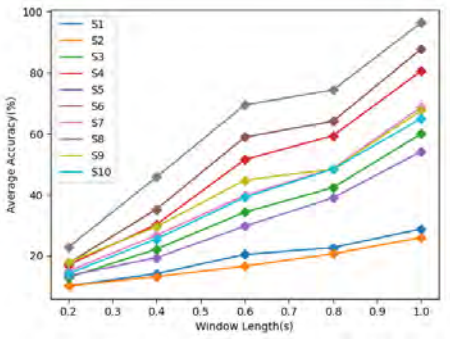
3.1.5 Results

Figure 8 compares the individual accuracies of all the 10 subjects by applying the CCA method using three different filters. The results vary tremendously among all the users. Subject 8 outperforms all the others across all the window lengths. Table 9 shows that subject 8 gains 96.5% accuracy when filtered by Butterworth while subject 2 only attains 26.1% accuracy at window length 1s. On the other hand, the accuracies of each subject from three different filters only have minor differences which means that filters only have little effect on the CCA method.

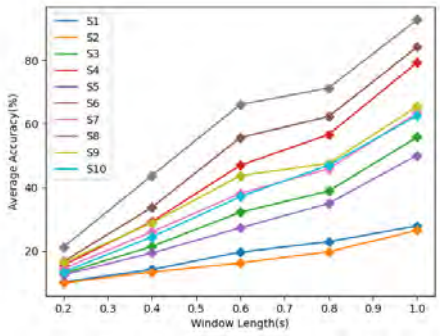
This can be justified by the average accuracies displayed in Figure 12(a) and Table 1.



(a) CCA with Butterworth Filter



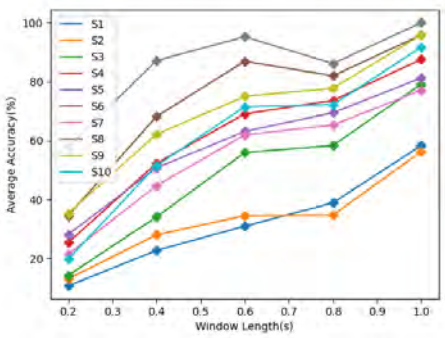
(b) CCA with Chebyshev Filter



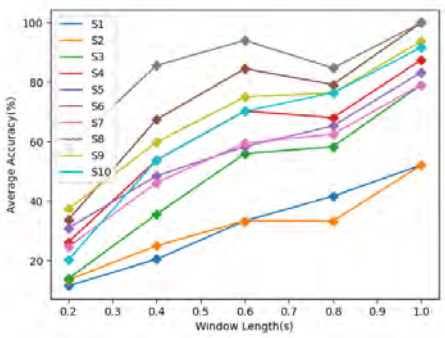
(c) CCA with Elliptic Filter

Figure 8: Results of CCA by using three different filters to preprocessing the data

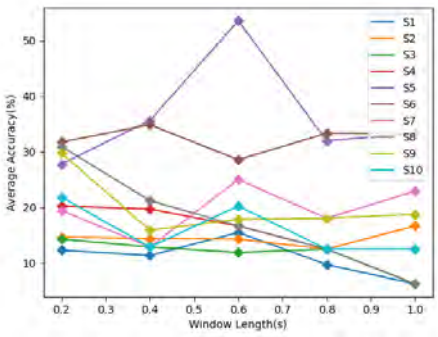
Figure 9 presents the results of 10 individual subjects of IT-CCA by applying filters Butterworth, Chebyshev, and Elliptic. For filters Butterworth and Chebyshev, subject 8 achieves the best accuracy while subjects 2 and 1 have the lowest accuracy. And the results vary little between the two different filters. However, for filter Elliptic, the results of individual subjects fluctuate at different window lengths and quite lower than filters Butterworth and Chebyshev.



(a) IT-CCA with Butterworth Filter



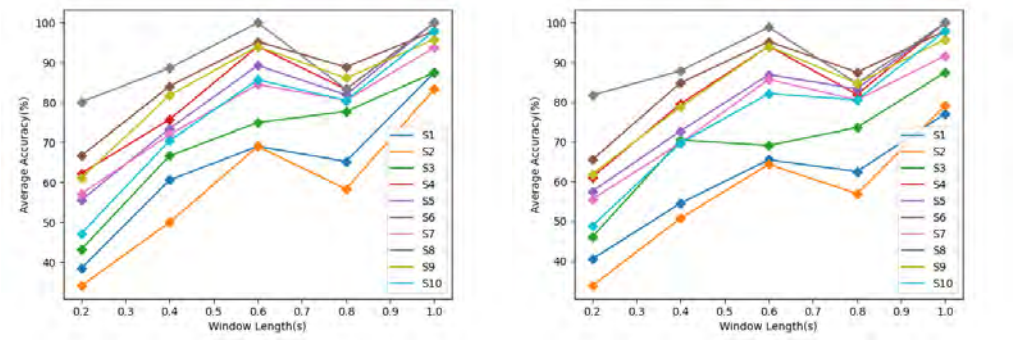
(b) IT-CCA with Chebyshev Filter



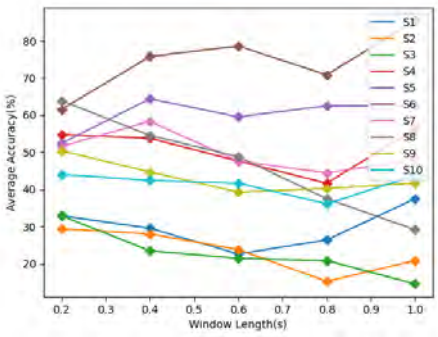
(c) IT-CCA with Elliptic Filter

Figure 9: Results of IT-CCA by using three different filters to preprocessing the data

Figure 10 displays the results of 10 subjects of the extended CCA method by the same three filters: Butterworth, Chebyshev, and Elliptic. Subject 8 still attains the best accuracy except at window length 0.8s by preprocessing with Butterworth and Chebyshev filters and subject 2 and 1 get the lowest accuracies. For filter Elliptic, subject 6 has the best result except at window length 0.2s.



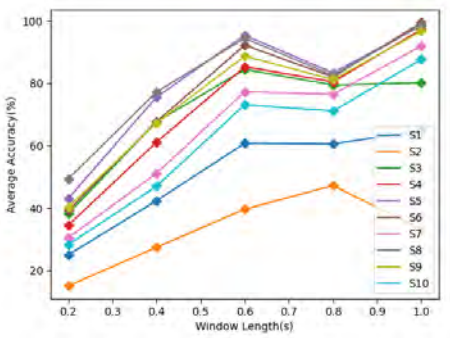
(a) Extended CCA with Butterworth Filter (b) Extended CCA with Chebyshev Filter



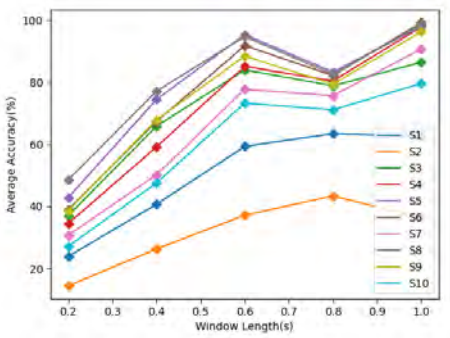
(c) Extended CCA with Elliptic Filter

Figure 10: Results of Extended CCA by using three different filters to preprocessing the data

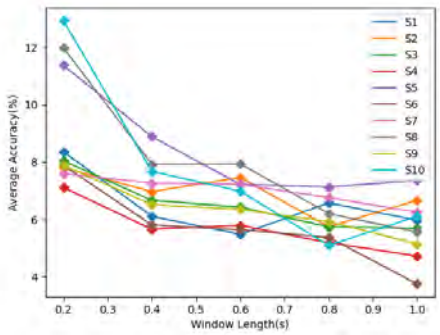
Figure 11 presents the results of the 10 subjects of C-CNN with the same three filters: Butterworth, Chebyshev, and Elliptic. Subject 5 and subject 8 gain the highest accuracy alternatively and there are not many differences between the two subjects. Subject 2 hast the lowest accuracy.



(a) C-CNN with Butterworth Filter



(b) C-CNN with Chebyshev Filter



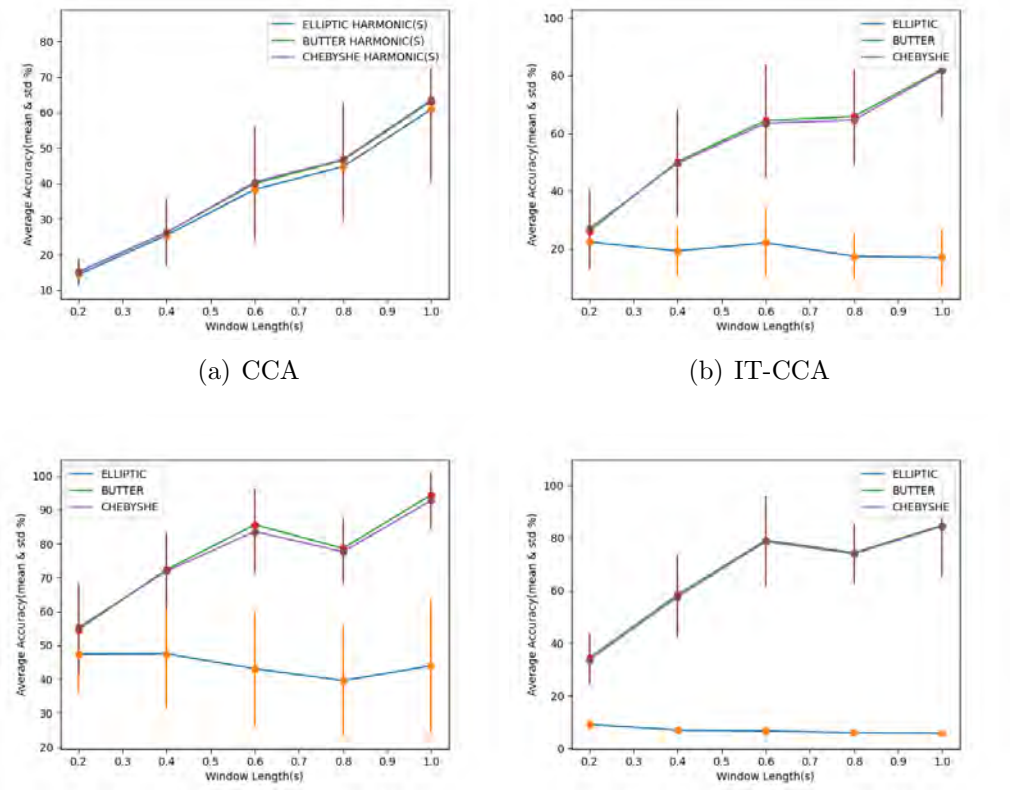
(c) C-CNN with Elliptic Filter

Figure 11: Results of C-CNN by using three different filters to preprocessing the data

Figure 12 and Table 1 present the average accuracies getting from CCA, IT-CCA, Extended CCA, and C-CNN by using Butterworth, Chebyshev, Elliptic Filters to preprocessing the EEG signals, respectively. The number of harmonics is set to 2 for all four methods and all three filters. As seen in Figure 12, for CCA, there is no much difference between the three filters except that results from the Elliptic filter are slightly lower than Butterworth and Chebyshev filters. However, for IT-CCA, Extended CCA, and C-CNN, results from Elliptic are dramatically lower than Butterworth and Chebyshev filters.

Take, subject 1, for example, Figure 13 compares the difference between the original EEG signals and the frequency domain representation of the same EEG singals of subject 1 with the filtered signals. Figure 13(b) indicates that noises in the original EEG signals dominate the frequency domain. After being filtered by Butterworth and Chebyshev, most noises are removed as shown in Figure 13(d) and 13(e) while after being filtered by Elliptic, the noises are not removed much as

shown in 13(h). This might explain why filters Butterworth and Chebyshev outperform Elliptic when using method IT-CCA, Extended CCA, and C-CNN since noises affect the calculation much more than standard CCA.
From table 1, for CCA, Butterworth slightly outperforms Chebyshev on window length 0.2s and 0.4s, but underperforms Chebyshev on window length 0.6s, 0.8s, and 1.0s. For IT-CCA, Chebyshev outperforms Butterworth only on window length 0.2s, and the same happens in Extended CCA. For C-CNN, Butterworth works better than Chebyshev on all window lengths, especially on window length 1.0s, Butterworth gains almost 5% higher accuracy than Chebyshev. Since Elliptic filter has a better fall-off but more ripple in pass-band and stop-band, Comparing with the Butterworth and the Chebyshev filters. We conclude that the fall-off is not critical when classify SSVEPS using CCA, IT-CCA, Extended CCA, and C-CNN. The Butterworth filter has the flattest pass-band and it outperforms the Chebyshev and Elliptic filters overall, so the pass-band contributes more than stop-band in methods CCA, IT-CCA, Extended CCA, and C-CNN.



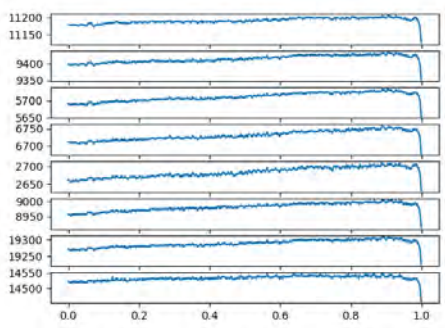
(a) CCA

(b) IT-CCA

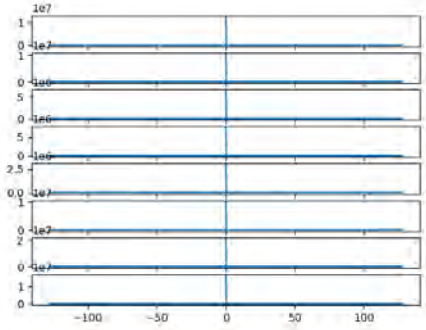
(c) Extended CCA

(d) C-CNN

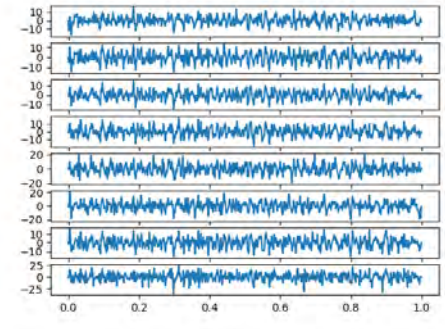
Figure 12: Results of CCA, IT-CCA, Extended CCA, and C-CCN by using three different filters to preprocessing the data



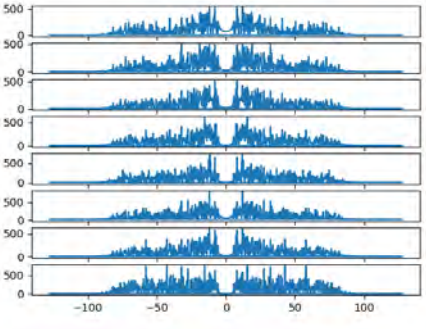
(a) Original Signals



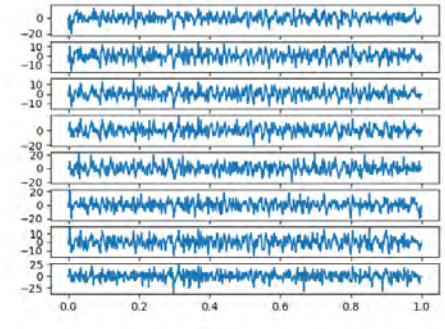
(b) Original Frequency Signals



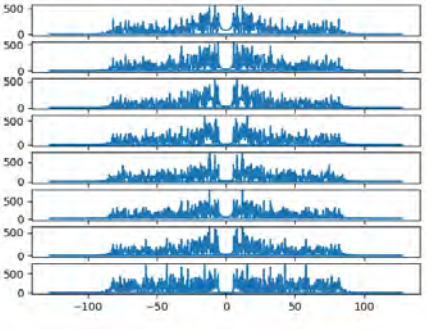
(c) EEG Signals of Butterworth



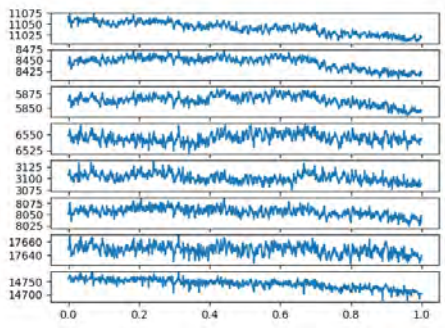
(d) Frequency Signals of Butterworth



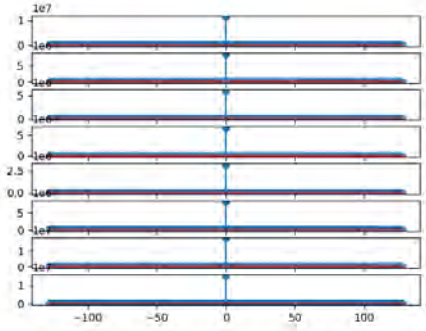
(e) EEG Signals of Chebyshev



(f) Frequency Signals of Chebyshev



(g) EEG Signals of Elliptic



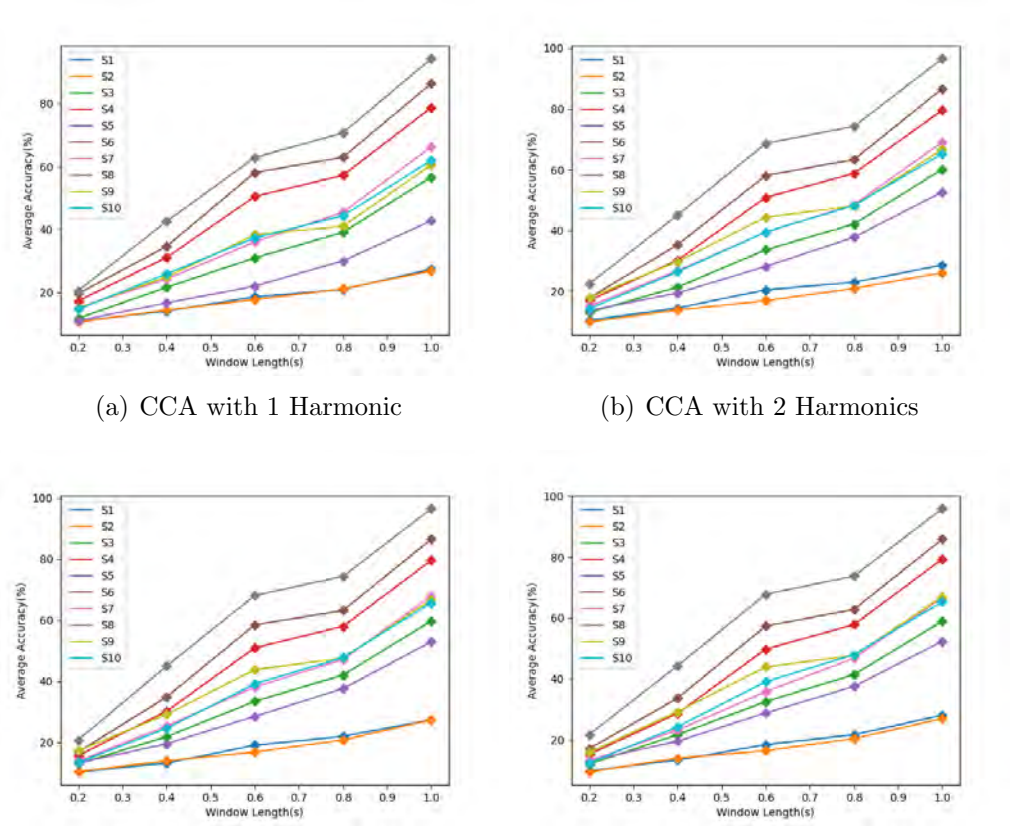
(h) Frequency Signals of Elliptic

Figure 13: EEG Signals and the Frequency presentation of the EEG Signals of Subject 1

		Window Length				
Method	Filter	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
CCA	Elliptic	14.4±3.2%	25.5±8.7%	38.3±14.7%	44.7±15.5%	60.8±20.8%
	Butterworth	15.2±3.6%	26.3±9.1%	40.0±15.4%	46.5±15.9%	63.1±21.6%
	Chebyshev	15.1±3.6%	26.2±9.3%	40.4±15.7%	46.8±16.0%	63.5±21.7%
IT-CCA	Elliptic	22.3±7.0%	19.2±8.6%	22.0±11.5%	17.4±8.0%	16.9±9.8%
	Butterworth	26.1±13.3%	50.2±18.4%	64.4±19.3%	65.8±16.4%	82.3±14.4%
	Chebyshev	27.0±13.3%	49.6±18.5%	63.5±18.8%	64.6±15.7%	81.9±16.5%
Extended CCA	Elliptic	47.4±11.5%	47.5±16.1%	43.1±17.0%	39.6±16.3%	44.0±19.7%
	Butterworth	54.6±13.3%	72.3±10.9%	85.6±10.6%	78.6±9.0%	94.4±5.8%
	Chebyshev	55.2±13.0%	71.9±11.4%	83.6±12.3%	77.6±9.7%	92.7±8.2%
C-CNN	Elliptic	9.1±2.0%	6.9±1.0%	6.7±0.8%	6.0±0.7%	5.7±1.0%
	Butterworth	34.4±9.4%	58.5±15.2%	79.2±16.6%	74.4±11.2%	84.8±20.0%
	Chebyshev	33.6±9.4%	57.6±15.4%	78.7±17.3%	74.0±11.7%	84.5±19.4%

Table 1: Results of CCA, IT-CCA, Extended CCA and C-CCN using preprocessing filters Elliptic, Butterworth and Chebyshev

After investigating the impacts of three filters Elliptic, Butterworth, and Chebyshev, we then experiment with the different numbers of harmonics to see how the harmonics affect methods CCA and Extended CCA. We use the Butterworth filter here to gain higher accuracy. As shown in Figure 16 and Table 2, the number of harmonics does not contribute much. This indicates that small changes in the target frequency signals contribute little to the final results. Although the difference between the results is very tiny, 2 harmonics outperforms the others and gains the highest average accuracy($mean \pm std$). The same pattern can be observed from the individual results as presented in Figures 14 and 15.



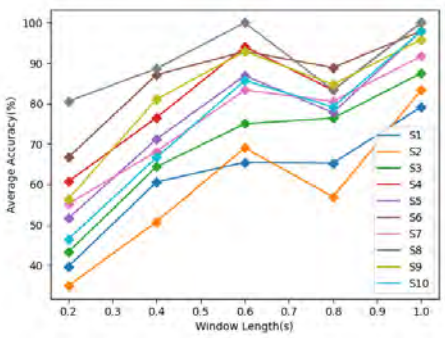
(a) CCA with 1 Harmonic

(b) CCA with 2 Harmonics

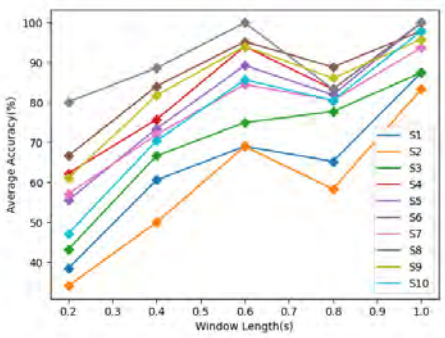
(c) CCA with 3 Harmonics

(d) CCA with 4 Harmonics

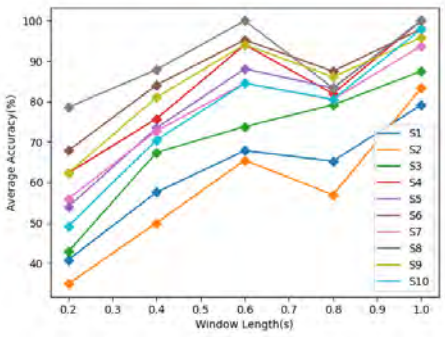
Figure 14: Individual Subject Results of CCA and 4 Different Number of Harmonics



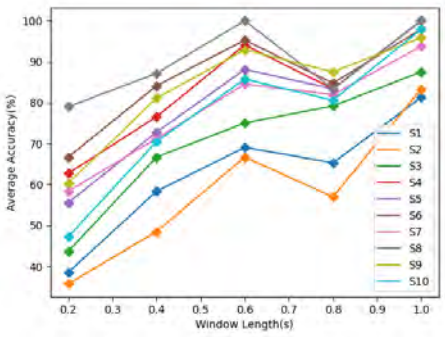
(a) Extended CCA Method with 1 Harmonic



(b) Extended CCA Method with 2 Harmonics

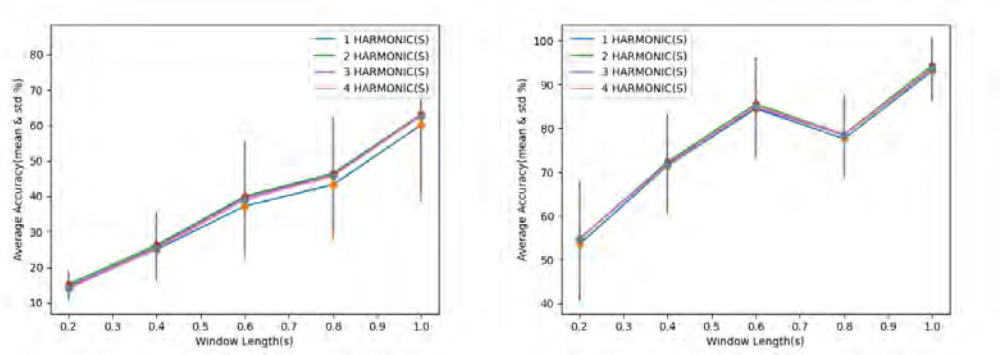


(c) Extended CCA Method with 3 Harmonics



(d) Extended CCA Method with 4 Harmonics

Figure 15: Individual Subject Results of Extended CCA Method and 4 Different Number of Harmonics



(a) CCA (b) Extended CCA

Figure 16: Results of CCA, Extended CCA by using 4 different number of harmonics

		Window Length				
Method	Number of Harmonics	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
CCA	1	14.6±3.4%	25.0±8.7%	37.2±15.1%	43.3±15.9%	60.2±21.8%
	2	15.2±3.6%	26.3±9.1%	40.0±15.4%	46.5±15.9%	63.1±21.6%
	3	14.5±3.1%	25.7±9.1%	39.6±15.6%	46.0±16.0%	63.0±21.6%
	4	14.0±3.5%	25.2±8.8%	39.0±15.5%	45.8±15.9%	62.6±21.3%
Extended CCA	1	53.5±12.9%	71.5±11.3%	84.5±10.8%	77.6±9.1%	93.1±7.0%
	2	54.6±13.3%	72.3±10.9%	85.6±10.6%	78.6±9.0%	94.4±5.8%
	3	54.9±12.7%	72.0±11.0%	84.8±11.4%	78.5±9.2%	93.5±7.2%
	4	54.8±12.8%	71.7±11.2%	85.1±10.8%	78.6±9.2%	93.8±6.8%

Table 2: Results of CCA, and Extended CCA using 1,2,3,4 number of harmonics

Figure 17 and Table 3 gives the averaged accuracy(mean ± std) across all subjects with different window lengths from 0.2s to 1s. The number of harmonics and filter are set to 2 and Butterworth, respectively. As shown in Figure 17, methods trained on individual subjects highly outperform standard CCA, in which Extended CCA wins. Overall the average accuracy increases monotonically from window length 0.2s to 1s for all methods, except Extended CCA and C-CNN drop on window length 0.8s which also duplicate in individual results of the 10 subjects as shown in Figure 18(c) and 18(c). During experiments, all the subjects are required to gaze at the targets without blinking or doing any other movements. Since the results drop at window length 0.8s, this might indicate that when users keep gazing at a

flicking target, they might be distracted or feel tired as time passing which might introduce noises to the EEG signals.
For the Extended CCA method and C-CNN, Figure **??**, Tables 5 and 7 present the detailed results of the 10 subjects where 9 of them decrease with window length 0.8s comparing with window length 0.6s. The reason might be that signals with window length 0.8s include more noises than window length 0.6s. On window length 0.2s, Extended CCA gains 54.6% average accuracy($mean \pm std$) which is dramatically higher than C-CNN which only has an accuracy of 34.4%. C-CNN slightly beats IT-CNN by 8.3%, followed by standard CCA which only gains 15.2%. For longer window lengths, Extended CCA defeats C_CNN from both average accuracy($mean \pm std$) and computational expense and it gains the highest average accuracy($mean \pm std$) 94.4% on window length 1.0s. Table 8 presents a detailed comparison.

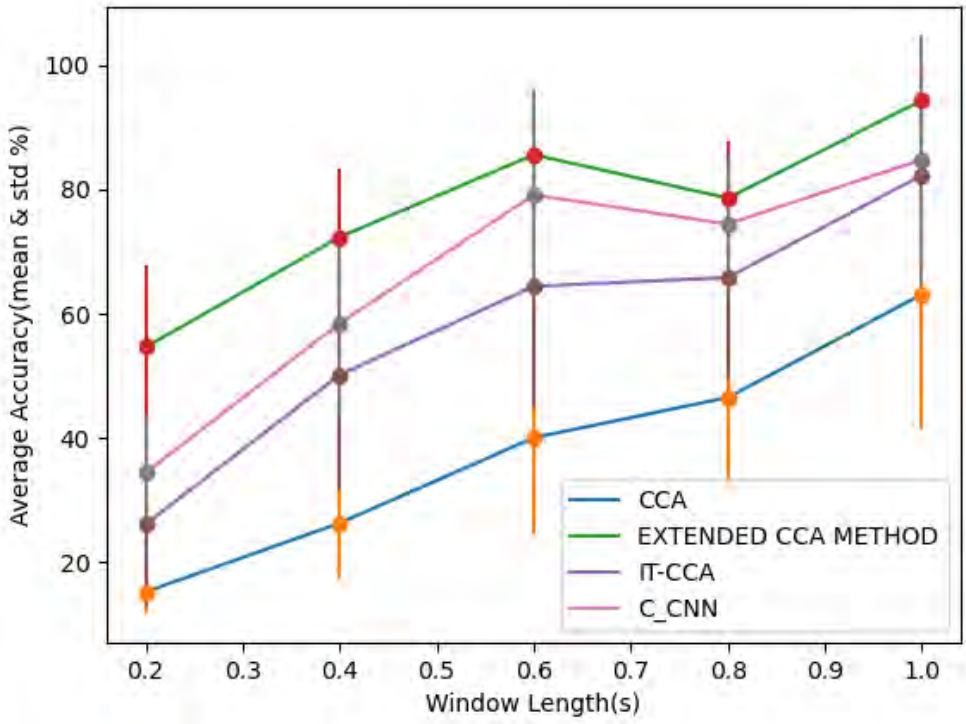
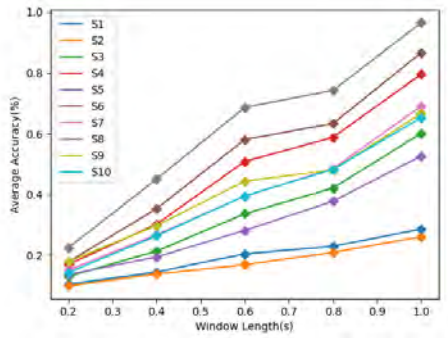


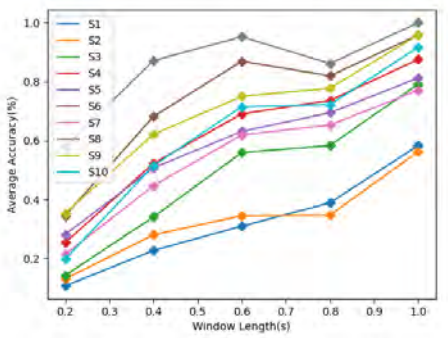
Figure 17: average accuracy($mean \pm std$) from CCA, Extended CCA, IT-CCA, and Complex Spectrum CNN.

	Window Length				
Method	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
CCA	15.2±3.6%	26.3±9.1%	40.0±15.4%	46.5±15.9%	63.1±21.6%
Extended CCA Method	54.6±13.3%	72.3±10.9%	85.6±10.6%	78.6±9.0%	94.4±5.8%
IT-CCA	26.1±13.3%	50.2±18.4%	64.4±19.3%	65.8±16.4%	82.3±14.4%
C_CNN	34.4±9.4%	58.5±15.2%	79.2±16.6%	74.4±11.2%	84.8±20.0%

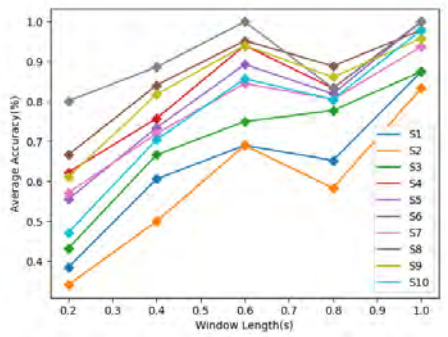
Table 3: Results of CCA, IT-CCA, Extended CCA, and C-CNN



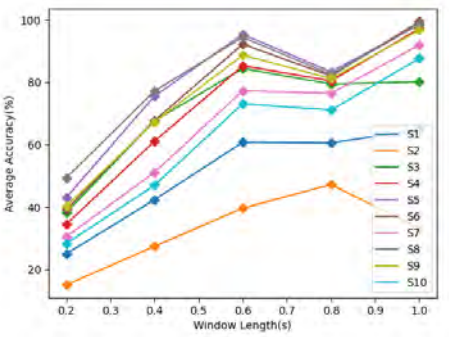
(a) CCA



(b) IT-CCA



(c) Extended CCA



(d) C-CNN

Figure 18: Individual Subject Results of CCA, IT-CCA, Extended CCA, and C-CCN

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.5%	14.5%	20.5%	23.0%	28.6%
S2	9.9%	13.8%	16.9%	20.9%	26.1%
S3	13.1%	21.4%	33.7%	42.1%	60.1%
S4	17.0%	30.2%	50.9%	58.8%	79.6%
S5	13.7%	19.4%	28.2%	37.8%	52.6%
S6	17.8%	35.3%	58.1%	63.3%	86.7%
S7	15.1%	26.8%	39.4%	48.5%	69.0%
S8	22.4%	45.1%	68.7%	74.3%	96.5%
S9	18.0%	29.7%	44.4%	48.1%	66.7%
S10	14.3%	26.4%	39.5%	48.2%	65.3%
mean ± std	15.2±3.6%	26.3±9.1%	40.0±15.4%	46.5±15.9%	63.1±21.6%

Table 4: Results of Individual Subject from CCA

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	38.5%	60.6%	69.0%	65.3%	87.5%
S2	34.1%	50.0%	69.0%	58.3%	83.3%
S3	43.3%	66.7%	75.0%	77.8%	87.5%
S4	62.3%	75.8%	94.0%	83.3%	100.0%
S5	55.6%	73.5%	89.3%	81.9%	100.0%
S6	66.7%	84.1%	95.2%	88.9%	97.9%
S7	57.1%	72.0%	84.5%	80.6%	93.8%
S8	80.2%	88.6%	100.0%	83.3%	100.0%
S9	61.1%	81.8%	94.0%	86.1%	95.8%
S10	47.2%	70.5%	85.7%	80.6%	97.9%
mean ± std	54.6±13.3%	72.3±10.9%	85.6±10.6%	78.6±9.0%	94.4±5.8%

Table 5: Results of Individual Subject from Extended CCA Method

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.7%	22.7%	31.0%	38.9%	58.3%
S2	13.1%	28.0%	34.5%	34.7%	56.2%
S3	14.3%	34.1%	56.0%	58.3%	79.2%
S4	25.4%	52.3%	69.0%	73.6%	87.5%
S5	28.2%	50.8%	63.1%	69.4%	81.2%
S6	34.5%	68.2%	86.9%	81.9%	95.8%
S7	21.4%	44.7%	61.9%	65.3%	77.1%
S8	57.9%	87.1%	95.2%	86.1%	100.0%
S9	35.3%	62.1%	75.0%	77.8%	95.8%
S10	19.8%	51.5%	71.4%	72.2%	91.7%
mean ± std	26.1±13.3%	50.2±18.4%	64.4±19.3%	65.8±16.4%	82.3±14.4%

Table 6: Results of Individual Subject from IT-CCA

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	25%	42.4%	60.8%	60.6%	64.9%
S2	15.1%	27.4%	39.7%	47.2%	33.3%
S3	38.2%	67.7%	84.4%	79.4%	80.1%
S4	34.5%	61.1%	85.3%	80.5%	97.2%
S5	43.1%	75.6%	95.4%	83.3%	98.5%
S6	39.2%	67.7%	92.2%	82.0%	99.4%
S7	30.5%	51.2%	77.3%	76.5%	91.9%
S8	49.4%	77.2%	94.4%	82.5%	98.6%
S9	40.3%	67.3%	88.7%	81.2%	96.7%
S10	28.3%	47.0%	73.2%	71.1%	87.6%
mean ± std	34.4±9.4%	58.5±15.2%	79.2±16.6%	74.4±11.2%	84.8±20.0%

Table 7: Results of Individual Subject from C-CNN

Method	Order of average accuracy($mean \pm std$)	Training	Order of Efficiency
Extended CCA Method	1	Yes	3
C_CNN	2	Yes	4
IT-CCA	3	Yes	2
CCA	4	No	1

Table 8: Comparison of CCA, IT-CCA, Extended CCA and C-CNN (The higher the number, the lower the order.)

4 Discussion and Future Work

In this study, we proved that subject-dependent training methods perform much better than standard CCA which requires no training. The quantity of pass-band of preprocessing filters contributes more to the final results than the ripple of stopband and the fall-off. The number of harmonics has no influence on the classification methods and the longer the window length, the more accurate the results.

The Extended CCA method gains $94.4 \pm 5.8\%$ average accuracy($mean \pm std$) in the offline dataset at the cost of longer calculation time which would extend the waiting time in online experiments. CNN using complex spectrum features of EEG signals as input also performs better than standard CCA. Comparing with Extended CCA and IT-CCA, the training process is much more complex and computationally inefficient. However, after pertaining to the best model, the evaluation process is much computational efficient than all the other three methods. Besides the four methods we present in this paper, there are still several other methods from the CCA family such as Filter Bank CCA(FBCCA), the cluster analysis of CCA coefficient(CACC), a phase constrained CCA (PCCA) method, etc. In the future, given time and resources, we plan to explore these methods, together with more CNN models.

Acknowledgment

I would like to express my sincere gratitude to my supervisor, Mohammad Davari, the CEO of InnoBrain, for providing me his constant support and advice during the period of my master's project. I would also like to thank Parivash Purabbasi and Shayan Jalilpour from Innobrain for directing me in the right direction when conducting experiments.
I would also like to thank professor Carolina Wählby, for willing to be my reviewer and providing valuable suggestions on my master thesis.
I would also like to thank professor Joakim Lindblad, for reviewing my project specification.
I would like to extend my gratitude to the coordinator of my previously taken course "Project in Computational Science", professor Maya Neytcheva and the supervisors, professor Natasa Sladoje and Joakim Lindblad for their valuable knowledge and advice about conducting academic research which is extremely helpful for my master project.
I would like to thank my parents, Mr. Yongjun Han and Ms. Changying Chen, for bringing me into the world and inspiring me through my life. I thank my brother, Qingzhong Han, for his encouragement and support.
I extend my gratitude to my friends, Rocky Zhang, Wei Peng, and Billy Chen for their accompanying and encouragement during this special time.

References

[1] Jessica Fertier. Introduction to filters: Fir versus iir. `https://community.sw.siemens.com/s/article/introduction-to-filters-fir-versus-iir`, 2020. [Online; accessed 6-October-2020].

[2] Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992.

[3] 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):e0140703, 2015.

[4] M. Nakanishi, Y. Wang, X. Chen, Y. Wang, X. Gao, and T. Jung. Enhancing detection of ssveps for a high-speed brain speller using task-related component analysis. *IEEE Transactions on Biomedical Engineering*, 65(1):104–112, 2018.

[5] Gilbert Strang. Introduction to linear algebra. In *Machine Learning, Dynamical Systems, and Control*, pages 490–496. Wellesley - Cambridge Press, 2016.

[6] J. Nathan Kutz Steven L. Brunton. Data-driven science and engineering. In *Machine Learning, Dynamical Systems, and Control*, pages 47–77. Cambridge University Press, 2019.

[7] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.

[8] Elizabeth Winter Wolpaw Jonathan R. Wolpaw. *BRAIN–COMPUTER INTERFACES*. Oxford University Press, Inc.

[9] Asanagi. Electrode locations of international 10-20 system for eeg (electroencephalography) recording. `https://commons.wikimedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg`, 2010. [Online; accessed 6-October-2020].

[10] Kian B Ng, Andrew P Bradley, and Ross Cunnington. Stimulus specificity of a steady-state visual-evoked potential-based brain–computer interface. *Journal of Neural engineering*, 9(3):036008, 2012.

[11] NR Galloway. Human brain electrophysiology: Evoked potentials and evoked magnetic fields in science and medicine. *The British journal of ophthalmology*, 74(4):255, 1990.

[12] Yee Joon Kim, Marcia Grabowecky, Ken A Paller, Krishnakumar Muthu, and Satoru Suzuki. Attention induces synchronization-based response gain in steady-state visual evoked potentials. *Nature neuroscience*, 10(1):117–125, 2007.

[13] Stephen Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

[14] Gregory Phelps. Real-time filtering in bioexplorer. *Brainquiry B.V*, 2018.

[15] Bruce Thompson. Canonical correlation analysis : Uses and interpretation sage university papers series. quantitative applications in the social sciences ; no. 07-047. 1984.

[16] 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, 2017.

[17] Francois Chollet. Deep learning with python. In *Machine Learning, Dynamical Systems, and Control*, page 87. Manning Publications Co., 2018.

Appendix

Individual Subject Results

Experiments with Filters

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.5%	14.5%	20.5%	23.0%	28.6%
S2	9.9%	13.8%	16.9%	20.9%	26.1%
S3	13.1%	21.4%	33.7%	42.1%	60.1%
S4	17.0%	30.2%	50.9%	58.8%	79.6%
S5	13.7%	19.4%	28.2%	37.8%	52.6%
S6	17.8%	35.3%	58.1%	63.3%	86.7%
S7	15.1%	26.8%	39.4%	48.5%	69.0%
S8	22.4%	45.1%	68.7%	74.3%	96.5%
S9	18.0%	29.7%	44.4%	48.1%	66.7%
S10	14.3%	26.4%	39.5%	48.2%	65.3%
mean ± std	15.2±3.6%	26.3±9.1%	40.0±15.4%	46.5±15.9%	63.1±21.6%

Table 9: Results of Individual Subject from CCA with Butterworth Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.1%	14.1%	20.4%	22.7%	28.7%
S2	10.2%	13.2%	16.6%	20.6%	26.0%
S3	12.9%	22.2%	34.3%	42.3%	60.0%
S4	17.0%	30.3%	51.5%	59.4%	80.6%
S5	13.5%	19.3%	29.7%	39.0%	54.2%
S6	17.5%	35.3%	58.9%	64.1%	87.8%
S7	14.9%	26.7%	39.8%	48.6%	68.8%
S8	22.7%	45.8%	69.4%	74.4%	96.4%
S9	17.9%	29.5%	44.8%	48.4%	67.6%
S10	14.1%	25.5%	39.1%	48.4%	65.0%
mean ± std	15.1±3.6%	26.2±9.3%	40.4±15.7%	46.8±16.0%	63.5±21.7%

Table 10: Results of Individual Subject from CCA with Chebyshev Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.3%	14.2%	19.7%	23.0%	27.9%
S2	10.0%	13.4%	16.3%	19.8%	26.5%
S3	13.0%	21.5%	32.2%	38.9%	55.8%
S4	15.5%	29.2%	47.0%	56.7%	79.3%
S5	12.6%	19.4%	27.3%	35.0%	50.1%
S6	16.7%	33.7%	55.7%	62.3%	84.2%
S7	14.5%	26.1%	38.2%	45.8%	63.5%
S8	21.3%	43.7%	66.1%	71.3%	92.8%
S9	16.4%	28.9%	43.7%	47.6%	65.4%
S10	13.4%	24.4%	37.1%	47.0%	62.6%
mean ± std	14.4±3.2%	25.5±8.7%	38.3±14.7%	44.7±15.5%	60.8±20.8%

Table 11: Results of Individual Subject from CCA with Elliptic Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.7%	22.7%	31.0%	38.9%	58.3%
S2	13.1%	28.0%	34.5%	34.7%	56.2%
S3	14.3%	34.1%	56.0%	58.3%	79.2%
S4	25.4%	52.3%	69.0%	73.6%	87.5%
S5	28.2%	50.8%	63.1%	69.4%	81.2%
S6	34.5%	68.2%	86.9%	81.9%	95.8%
S7	21.4%	44.7%	61.9%	65.3%	77.1%
S8	57.9%	87.1%	95.2%	86.1%	100.0%
S9	35.3%	62.1%	75.0%	77.8%	95.8%
S10	19.8%	51.5%	71.4%	72.2%	91.7%
mean ± std	26.1±13.3%	50.2±18.4%	64.4±19.3%	65.8±16.4%	82.3±14.4%

Table 12: Results of Individual Subject from IT-CCA with Butterworth Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	11.5%	20.5%	33.3%	41.7%	52.1%
S2	13.5%	25.0%	33.3%	33.3%	52.1%
S3	13.9%	35.6%	56.0%	58.3%	79.2%
S4	26.2%	53.8%	70.2%	68.1%	87.5%
S5	31.0%	48.5%	58.3%	65.3%	83.3%
S6	33.7%	67.4%	84.5%	79.2%	100.0%
S7	24.6%	46.2%	59.5%	62.5%	79.2%
S8	57.9%	85.6%	94.0%	84.7%	100.0%
S9	37.3%	59.8%	75.0%	76.4%	93.8%
S10	20.2%	53.8%	70.2%	76.4%	91.7%
mean ± std	27.0±13.3%	49.6±18.5%	63.5±18.8%	64.6±15.7%	81.9±16.5%

Table 13: Results of Individual Subject from IT-CCA with Chebyshev Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	12.3%	11.4%	15.5%	9.7%	6.2%
S2	14.7%	14.4%	14.3%	12.5%	16.7%
S3	14.3%	12.9%	11.9%	12.5%	6.2%
S4	20.2%	19.7%	16.7%	12.5%	12.5%
S5	27.8%	35.6%	53.6%	31.9%	33.3%
S6	31.7%	34.8%	28.6%	33.3%	33.3%
S7	19.4%	12.9%	25.0%	18.1%	22.9%
S8	31.0%	21.2%	16.7%	12.5%	6.2%
S9	29.8%	15.9%	17.9%	18.1%	18.8%
S10	21.8%	12.9%	20.2%	12.5%	12.5%
mean ± std	22.3±7.0%	19.2±8.6%	22.0±11.5%	17.4±8.0%	16.9±9.8%

Table 14: Results of Individual Subject from IT-CCA with Elliptic Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	38.5%	60.6%	69.0%	65.3%	87.5%
S2	34.1%	50.0%	69.0%	58.3%	83.3%
S3	43.3%	66.7%	75.0%	77.8%	87.5%
S4	62.3%	75.8%	94.0%	83.3%	100.0%
S5	55.6%	73.5%	89.3%	81.9%	100.0%
S6	66.7%	84.1%	95.2%	88.9%	97.9%
S7	57.1%	72.0%	84.5%	80.6%	93.8%
S8	80.2%	88.6%	100.0%	83.3%	100.0%
S9	61.1%	81.8%	94.0%	86.1%	95.8%
S10	47.2%	70.5%	85.7%	80.6%	97.9%
mean ± std	54.6±13.3%	72.3±10.9%	85.6±10.6%	78.6±9.0%	94.4±5.8%

Table 15: Results of Individual Subject from Extended CCA with Butterworth Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	40.5%	54.5%	65.5%	62.5%	77.1%
S2	33.7%	50.8%	64.3%	56.9%	79.2%
S3	46.0%	70.5%	69.0%	73.6%	87.5%
S4	61.1%	79.5%	94.0%	81.9%	100.0%
S5	57.5%	72.7%	86.9%	83.3%	100.0%
S6	65.5%	84.8%	95.2%	87.5%	97.9%
S7	55.6%	69.7%	85.7%	80.6%	91.7%
S8	81.7%	87.9%	98.8%	84.7%	100.0%
S9	61.9%	78.8%	94.0%	84.7%	95.8%
S10	48.8%	69.7%	82.1%	80.6%	97.9%
mean ± std	55.2±13.0%	71.9±11.4%	83.6±12.3%	77.6±9.7%	92.7±8.2%

Table 16: Results of Individual Subject from Extended CCA with Chebyshev Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	32.9%	29.5%	22.6%	26.4%	37.5%
S2	29.4%	28.0%	23.8%	15.3%	20.8%
S3	32.9%	23.5%	21.4%	20.8%	14.6%
S4	54.8%	53.8%	47.6%	41.7%	56.2%
S5	52.4%	64.4%	59.5%	62.5%	62.5%
S6	61.5%	75.8%	78.6%	70.8%	85.4%
S7	51.6%	58.3%	47.6%	44.4%	47.9%
S8	63.9%	54.5%	48.8%	37.5%	29.2%
S9	50.4%	44.7%	39.3%	40.3%	41.7%
S10	44.0%	42.4%	41.7%	36.1%	43.8%
mean ± std	47.4±11.5%	47.5±16.1%	43.1±17.0%	39.6±16.3%	44.0±19.7%

Table 17: Results of Individual Subject from Extended CCA with Elliptic Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	25.0%	42.4%	60.8%	60.6%	64.9%
S2	15.1%	27.4%	39.7%	47.2%	33.3%
S3	38.2%	67.7%	84.4%	79.4%	80.1%
S4	34.5%	61.1%	85.3%	80.5%	97.2%
S5	43.1%	75.6%	95.4%	83.3%	98.5%
S6	39.2%	67.7%	92.2%	82.0%	99.4%
S7	30.5%	51.2%	77.3%	76.5%	91.9%
S8	49.4%	77.2%	94.4%	82.5%	98.6%
S9	40.3%	67.3%	88.7%	81.2%	96.7%
S10	28.3%	47.0%	73.2%	71.1%	87.6%
mean ± std	34.4±9.4%	58.5±15.2%	79.2±16.6%	74.4±11.2%	84.8±20.0%

Table 18: Results of Individual Subject from C_CNN with Butterworth Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	23.8%	40.7%	59.3%	63.4%	62.5%
S2	14.4%	26.3%	37.1%	43.3%	36.2%
S3	37.0%	65.8%	84.0%	78.8%	86.4%
S4	34.4%	59.1%	85.1%	80.4%	97.8%
S5	42.8%	74.4%	95.3%	83.2%	97.9%
S6	38.9%	67.3%	91.7%	82.1%	99.2%
S7	30.7%	50.2%	77.7%	75.6%	90.7%
S8	48.5%	77.1%	94.7%	82.5%	98.5%
S9	38.5%	67.8%	88.4%	79.4%	96.2%
S10	27.1%	47.5%	73.3%	71.1%	79.6%
mean ± std	33.6±9.4%	57.6±15.4%	78.7±17.3%	74.0±11.7%	84.5±19.4%

Table 19: Results of Individual Subject from C_CNN with Chebyshev Filter

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	8.4%	6.1%	5.5%	6.6%	6.0%
S2	7.8%	7.0%	7.5%	5.7%	6.7%
S3	8.0%	6.7%	6.4%	5.7%	5.7%
S4	7.1%	5.7%	5.8%	5.2%	4.7%
S5	11.4%	8.9%	7.2%	7.1%	7.4%
S6	7.9%	5.8%	5.6%	5.4%	3.8%
S7	7.6%	7.3%	7.2%	6.8%	6.3%
S8	12.0%	7.9%	7.9%	6.2%	5.6%
S9	7.9%	6.5%	6.3%	5.9%	5.1%
S10	12.9%	7.7%	7.0%	5.1%	6.1%
mean ± std	9.1±2.0%	6.9±1.0%	6.7±0.8%	6.0±0.7%	5.7±1.0%

Table 20: Results of Individual Subject from C_CNN with Elliptic Filter

Experiments with Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.8%	14.0%	18.6%	20.9%	27.4%
S2	10.6%	14.3%	17.7%	21.1%	26.8%
S3	11.8%	21.6%	31.0%	39.1%	56.7%
S4	17.3%	31.1%	50.5%	57.2%	78.8%
S5	11.0%	16.6%	22.0%	30.0%	42.8%
S6	19.6%	34.6%	58.1%	63.0%	86.4%
S7	15.2%	24.2%	36.0%	45.5%	66.2%
S8	20.5%	42.5%	62.9%	70.7%	94.3%
S9	15.0%	24.8%	38.3%	41.0%	60.6%
S10	14.7%	25.9%	37.3%	44.4%	61.9%
mean ± std	14.6±3.4%	25.0±8.7%	37.2±15.1%	43.3±15.9%	60.2±21.8%

Table 21: Results of Individual Subject from CCA with 1 Harmonic

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.5%	14.5%	20.5%	23.0%	28.6%
S2	9.9%	13.8%	16.9%	20.9%	26.1%
S3	13.1%	21.4%	33.7%	42.1%	60.1%
S4	17.0%	30.2%	50.9%	58.8%	79.6%
S5	13.7%	19.4%	28.2%	37.8%	52.6%
S6	17.8%	35.3%	58.1%	63.3%	86.7%
S7	15.1%	26.8%	39.4%	48.5%	69.0%
S8	22.4%	45.1%	68.7%	74.3%	96.5%
S9	18.0%	29.7%	44.4%	48.1%	66.7%
S10	14.3%	26.4%	39.5%	48.2%	65.3%
mean ± std	15.2±3.6%	26.3±9.1%	40.0±15.4%	46.5±15.9%	63.1±21.6%

Table 22: Results of Individual Subject from CCA with 2 Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	10.3%	13.2%	19.0%	21.9%	27.4%
S2	10.4%	13.9%	16.8%	20.7%	27.4%
S3	13.0%	21.8%	33.4%	42.0%	59.7%
S4	15.6%	30.2%	51.0%	58.0%	79.6%
S5	13.2%	19.5%	28.5%	37.6%	52.9%
S6	16.9%	34.8%	58.5%	63.1%	86.5%
S7	13.9%	25.3%	38.1%	47.1%	67.9%
S8	20.8%	44.9%	68.1%	74.4%	96.5%
S9	17.2%	29.1%	43.8%	47.6%	66.7%
S10	13.4%	24.5%	39.1%	47.8%	65.7%
mean ± std	14.5±3.1%	25.7±9.1%	39.6±15.6%	46.0±16.0%	63.0±21.6%

Table 23: Results of Individual Subject from CCA with 3 Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	9.8%	13.3%	18.4%	21.7%	28.1%
S2	9.5%	14.0%	16.4%	20.4%	26.9%
S3	12.0%	21.7%	32.5%	41.4%	58.9%
S4	15.2%	28.6%	49.8%	57.8%	79.3%
S5	13.2%	19.5%	28.8%	37.5%	52.4%
S6	17.1%	33.6%	57.3%	62.8%	85.8%
S7	13.1%	23.2%	35.8%	46.8%	66.7%
S8	21.7%	44.3%	67.8%	73.7%	95.8%
S9	15.9%	29.2%	43.9%	47.7%	66.9%
S10	12.4%	24.2%	39.0%	48.0%	65.3%
mean ± std	14.0±3.5%	25.2±8.8%	39.0±15.5%	45.8±15.9%	62.6±21.3%

Table 24: Results of Individual Subject from CCA with 4 Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	39.7%	60.6%	65.5%	65.3%	79.2%
S2	34.9%	50.8%	69.0%	56.9%	83.3%
S3	43.3%	64.4%	75.0%	76.4%	87.5%
S4	60.7%	76.5%	94.0%	83.3%	100.0%
S5	51.6%	71.2%	86.9%	77.8%	97.9%
S6	66.7%	87.1%	92.9%	88.9%	97.9%
S7	55.2%	68.2%	83.3%	80.6%	91.7%
S8	80.6%	88.6%	100.0%	83.3%	100.0%
S9	56.3%	81.1%	92.9%	84.7%	95.8%
S10	46.4%	66.7%	85.7%	79.2%	97.9%
mean ± std	53.5±12.9%	71.5±11.3%	84.5±10.8%	77.6±9.1%	93.1±7.0%

Table 25: Results of Individual Subject from Extended CCA method with 1 Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	38.5%	60.6%	69.0%	65.3%	87.5%
S2	34.1%	50.0%	69.0%	58.3%	83.3%
S3	43.3%	66.7%	75.0%	77.8%	87.5%
S4	62.3%	75.8%	94.0%	83.3%	100.0%
S5	55.6%	73.5%	89.3%	81.9%	100.0%
S6	66.7%	84.1%	95.2%	88.9%	97.9%
S7	57.1%	72.0%	84.5%	80.6%	93.8%
S8	80.2%	88.6%	100.0%	83.3%	100.0%
S9	61.1%	81.8%	94.0%	86.1%	95.8%
S10	47.2%	70.5%	85.7%	80.6%	97.9%
mean ± std	26.1±54.6± 13.3%	72.3±10.9%	85.6±10.6%	78.6±9.0%	94.4±5.8%

Table 26: Results of Individual Subject from Extended CCA method with 2 Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	40.9%	57.6%	67.9%	65.3%	79.2%
S2	34.9%	50.0%	65.5%	56.9%	83.3%
S3	42.9%	67.4%	73.8%	79.2%	87.5%
S4	62.3%	75.8%	94.0%	81.9%	100.0%
S5	54.0%	73.5%	88.1%	83.3%	100.0%
S6	67.9%	84.1%	95.2%	87.5%	97.9%
S7	56.0%	72.7%	84.5%	80.6%	93.8%
S8	78.6%	87.9%	100.0%	83.3%	100.0%
S9	62.3%	81.1%	94.0%	86.1%	95.8%
S10	49.2%	70.5%	84.5%	80.6%	97.9%
mean ± std	54.9±12.7%	72.0±11.0%	84.8±11.4%	78.5±9.2%	93.5±7.2%

Table 27: Results of Individual Subject from Extended CCA method with 3 Harmonics

	Window Length				
Subject	0.2(s)	0.4(s)	0.6(s)	0.8(s)	1.0(s)
S1	38.5%	58.3%	69.0%	65.3%	81.2%
S2	35.7%	48.5%	66.7%	56.9%	83.3%
S3	43.7%	66.7%	75.0%	79.2%	87.5%
S4	62.7%	76.5%	94.0%	83.3%	100.0%
S5	55.6%	72.7%	88.1%	83.3%	100.0%
S6	66.7%	84.1%	95.2%	84.7%	97.9%
S7	58.3%	71.2%	84.5%	81.9%	93.8%
S8	79.0%	87.1%	100.0%	83.3%	100.0%
S9	60.3%	81.1%	92.9%	87.5%	95.8%
S10	47.2%	70.5%	85.7%	80.6%	97.9%
mean ± std	54.8±12.8%	71.7±11.2%	85.1±10.8%	78.6±9.2%	93.8±6.8%

Table 28: Results of Individual Subject from Extended CCA method with 4 Harmonics