Complex CNN incorporating Hilbert transform for steady-state visual evoked potential BCI

Rintaro Takata* and Yoshikazu Washizawa†
* University of Electro-Communications, Tokyo
E-mail: t2331089@edu.cc.uec.ac.jp Tel: +81-0424435287
† University of Electro-Communications, Tokyo
E-mail: washizawa@uec.ac.jp Tel: +81-0424435976

***Abstract*—Brain computer interface (BCI) is a system that converts brain activity information into commands, allowing people to communicate with the outside world without the need for physical activity. Steady-state visual evoked potential (SSVEP) which is a kind of event related potentials (ERPs) is used for BCI. SSVEP shows higher performance and has been improved in various ways to improve the information transfer rate and expand the number of commands, such as phase-modulated SSVEP-BCI. To further improve the performance of SSVEP-BCI, the frequency and phase of SSVEP need to be detected in a short time.**

In this study, we apply the Hilbert transform to the complex-valued convolutional neural network (CVCNN) for phase-modulated SSVEP-BCI. The Hilbert transform and accompanying analytic signal enable us to determine the instantaneous frequency and phase. The proposed CVCNN incorporating the Hilbert transform not only detects instantaneous frequency and phase information from electroencephalogram (EEG), but also learns convolutoinal filter automatically to detect SSVEP frequency, phase, and their harmonics components. As the result of evaluation experiments on open SSVEP datasets, the proposed method showed higher accuracy than conventional methods.

I. Introduction

Brain-computer interface (BCI) is a system to operate computers or communicate with the outside world without the need for physical activity [1]. BCI is expected to be used as a communication tool for people who cannot move their muscles freely, such as patients with amyotrophic lateral sclerosis (ALS). Futhermore, BCI is applied for healthy people in areas such as gaming, entertainment, security, and education [2].

Electroencephalogram (EEG) is a scalp-based measurement of changes in electrical potentials due to cranial nerve activity. Event related potentials (ERPs) occurs in association with perceptual and cognitive processing. Visual evoked potential (VEP) is a kind of ERPs, that is evoked by visual stimuli. Steady-state visual evoked potential (SSVEP) is a kind of VEP, that is elicited by periodic flashes. SSVEP has higher signal-to-noise ratio than the other ERPs and less individual differences, thus, is applied to BCI. SSVEP-BCI has shown high accuracy [3], [4].

SSVEP is easily evoked around 10 Hz, and there is a limit to the frequency used for visual stimulation. The phase-modulated SSVEP was devised to increase the number of commands. The phase-modulated SSVEP attempts to modulate not only by frequency but also phase information. Extended canonical correlation analysis (CCA) method was proposed to classify the phase-modulated SSVEP, and showed high information transfer rate (ITR) of 117.75 bits/min and 105 bits/min [5], [6].

In recent years, deep learning has attracted much attention due to its success in pattern recognition such as image classification, object detection, and natural language processing. These are due to the easy availability of large datasets and increased computing power. Convolutional neural network (CNN) is one particularly successful model, which extracts features by convolution to provide position invariance to the model [7]–[9].

A complex-valued neural network (CVNN) is a neural network model that extends weights, inputs, outputs, activation functions, etc. to complex numbers. CVNN has been performed better than real-valued neural network (RVNN) in various fields, because complex numbers are suitable for representing amplitude and phase to process in the frequency domain via Fourier transforms, etc. [10]–[13].

This study proposes a complex valued convolutional neural network (CVCNN) model that incorporates the Hilbert transform. By the Hilbert transform, the proposed neural network model generates an complex analytic signal within the neural network model to obtain instantaneous amplitude and phase. We used this model for phase-modulated SSVEP classification. By extracting instantaneous amplitude and phase in the neural network, phase-modulated SSVEP is detected from short time frame accurately. We compare the performance of the proposed model with conventional methods and typical CNNs.

II. Method

A. Hilbert transform and analytic signal

The sinusoidal signal $x(t)$, whose amplitude and frequency vary with time, is given by

$$\begin{aligned} x(t) &= A(t)\cos(\omega(t)t+\theta) \\ &= A(t)\cos(\phi(t)), \quad \phi(t)=\omega(t)t+\theta, \end{aligned} \tag{1}$$

where $A(t)$ is the instantaneous amplitude, $\omega(t)$ is the instantaneous angular frequency, and θ is the initial phase. In order to determine the instantaneous amplitude $A(t)$ and frequency

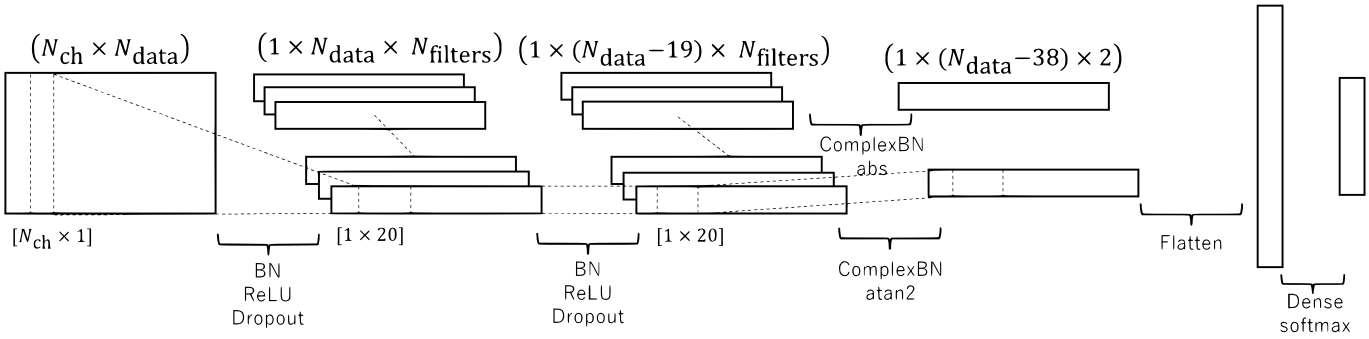


Fig. 1: Structure of the proposed network model

$\omega(t)$, the sinusoidal signal $x(t)$ is extended to complex analytic signal by the Hilbert transform.

The Hilbert transform shifts the phase of the sinusoidal signal $x(t)$ by 90 degrees. The Hilbert transform $y(t)$ of $x(t)$ is defined by the convolution of $x(t)$ and $1/\pi t$,

$$y(t) = x(t) * \frac{1}{\pi t} = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau. \tag{2}$$

The analytic signal $z(t)$ is generated by putting the signal $x(t)$ in the real part and the Hilbert transform $y(t)$ in the imaginary part,

$$\begin{aligned} z(t) &= x(t) + jy(t) \\ &= A(t)\cos(\phi(t)) + jA(t)\sin(\phi(t)) \\ &= A(t)e^{j\phi(t)}. \end{aligned} \tag{3}$$

The instantaneous amplitude $A(t)$ and instantaneous phase $\phi(t)$ are respectively obtained as follows,

$$A(t) = \sqrt{x(t)^2 + y(t)^2} \tag{4}$$

$$\phi(t) = \mathrm{atan2}(y(t), x(t)), \tag{5}$$

where atan2 function calculates the declination angle in the complex plane. In general, the instantaneous angular frequency $\omega(t)$ is obtained by the derivative of $\phi(t)$.

The Hilbert transform filter is an FIR filter that performs the Hilbert transform (2). Its coefficients is generated by Parks-McClellan algorithm. The output signal is delayed in proportion to the size of the FIR filter. When applying an M-dimensional Hilbert transform filter, the signal is delayed by $\frac{M}{2}$ sampling points.

B. Proposed neural network

The proposed neural network achieves instantaneous frequency analysis through the Hilbert transform in the convolutional layer. The complex analytic signal is generated by the convolutional layer, and the latter complex valued activation function extracts frequency and amplitude information.

The activation function of the complex valued layer is not as simple as the real valued network, because it is a complex function. In the conventional CVNN, $\mathrm{ReLU}(x(t)) + i\mathrm{ReLU}(y(t))$ or $\tanh(|z(t)|)\exp(j\arg z(t))$ is used [14]–[16]. In this research, we use the absolute value function $|z(t)| = \sqrt{x(t)^2 + y(t)^2}$ and the declination function $\arg z(t) = \mathrm{atan2}(y(t), x(t))$ to obtain instantaneous amplitude and phase.

The structure of the proposed network model is shown in Fig. 1. BN and ComplexBN refer to batch normalization and complex batch normalization, respectively. This network model consists of three convolutional layers and one fully connected layer. The size of input matrix is the number of samplings N_{data} multiplied by the number of channels N_{ch}. The size of filters is N_{filters}.

The first convolutional layer reduces dimensionality with respect to the channel direction. After the convolution, batch normalization, activation function ReLU, and the dropout are applied.

In the second convolutional layer, the convolution is performed with respect to the time direction. This convolutional layer generates narrowband signals corresponding to the stimulus frequencies and harmonics of SSVEP.

The third convolution layer generates analytic signal. After the convolution, the absolute value function and the atan2 function are applied as the activation functions. Then the values are flattened, and discriminated by the fully connected layer. We set the initial value of the convolution filter in the third layer to the Hilbert transform filter. We considered two cases where the filter weights are updated and fixed during the learning process. We refer to them as "proposed train" and "proposed untrain" respectively.

III. Experiment

A. Datasets

We used two open datasets. The first dataset is phase-modulated SSVEP data of 35 subjects with 40 classes [5]. We used the same nine channels and eight experimental subjects as [5]. The sampling frequency was 250 Hz. The visual stimuli consist of 40 frequencies ranging from 8 to 15.8 Hz at 0.2 Hz intervals. The initial phase of each stimulus is also shifted by 0.5π. For each subject and each stimulus, six trials of data were collected. Similar to the reference study, the data were divided into training and test sets using six-fold cross-validation [5].

The second dataset is phase-modulated SSVEP data from ten subjects in twelve targets [6]. The following method is

used to create a series of visual stimuli at all frequencies and initial phases regardless of the refresh rate.

Let f be the frequency of the steady blinking stimulus, ϕ be the initial phase and N_f be the refresh rate. The visual stimulus is generated as follows,

$$s(f, \phi, n) = \text{square}[2\pi f(n/N_f) + \phi], \tag{6}$$

where square[] generates a 50% duty cycle square wave with levels 0 and 1. The stimulus sequence $s_i(n)$ of the i-th targets is defined as

$$s_i(n) = s(f_0 + (i-1)\Delta f, \phi_0 + (i-1)\Delta\phi, n), \\ i = 1, 2, \ldots, N_{\text{targets}}, \tag{7}$$

where f_0 is the minimum frequency used for stimulation, ϕ_0 is the initial phase at frequency f_0, Δf and $\Delta\phi$ are the frequency and phase intervals respectively, and $N_{\text{targets}} = 12$ is the number of frequencies used for stimulation [6]. The twelve visual stimuli were presented at a refresh rate of 60 Hz, and the frequency and phase values were $(f_0 = 9.25\text{Hz}, \Delta f = 0.5\text{Hz})$ and $(\phi_0 = 0, \Delta\phi = 0.5\pi)$. Sampling frequency was 256 Hz. For each subject and each stimulus, fifteen trials of data were collected. Similar to the reference study, the data were divided into training and test sets using fifteen-fold cross-validation [6].

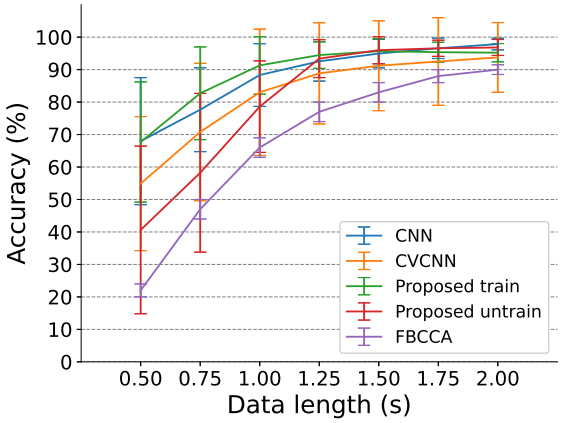
B. Methods

We compared three network models: real-valued convolutional neural network (RVCNN), CVCNN, and the proposed method. In the following, we will refer to RVCNN as CNN. For CNN and CVCNN, we used models similar to the structure of the proposed method. The both models have two convolutional layers and one fully connected layer. The number of filters and filter size were set to the same values as for the proposed method. For CNN, ReLU was used as the activation function after the convolutional layer, and the softmax function was placed after the fully connected layer. The activation functions of CVCNN are as follows; $\tanh(|z|)\exp(j\arg z)$ is used after the first convolutional layer, $\sqrt{x(t)^2 + y(t)^2}$ is used after the complex fully connected layer. In CVCNN, complex batch normalization was applied for the batch normalization part [15].

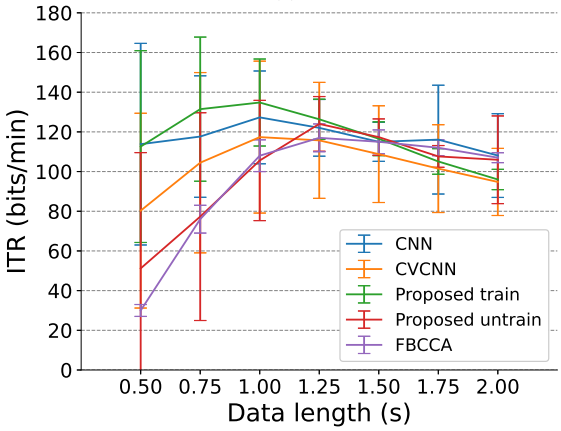
Grid searches were executed for the parameter selection of L2 regularization, dropout rate, learning rate, and number of epochs in $\{0.0001, 0.001, 0.01\}$, $\{0.0, 0.2, 0.4\}$, $\{0.0001, 0.001, 0.01\}$, $\{300, 400, 500\}$, respectively. Adam was used for optimizer. The loss function is mean squared error. All neural network models and experiments were implemented using Python 3.11.3, Numpy 1.24.3, Keras 2.13.1, and Tensorflow 2.13.0. We used remez function of the Python library Scipy for the Hilbert transform filter.

We conducted a comparison based on classification accuracy and information transfer rate (ITR). ITR represents the amount of information that can be transmitted per unit time. Let N be the number of targets, P be the classification accuracy, and T be the data length. ITR was calculated as follows,

$$\frac{1}{T}(\log_2 N + P\log_2 P + (1-P)\log_2(\frac{1-P}{N-1})) \text{ [bits/min]}. \tag{8}$$



(a)

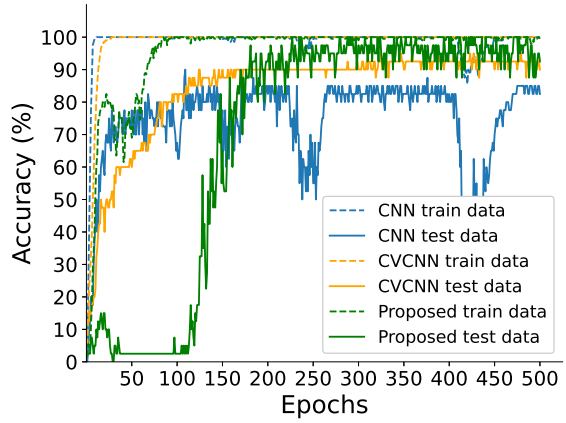


(b)

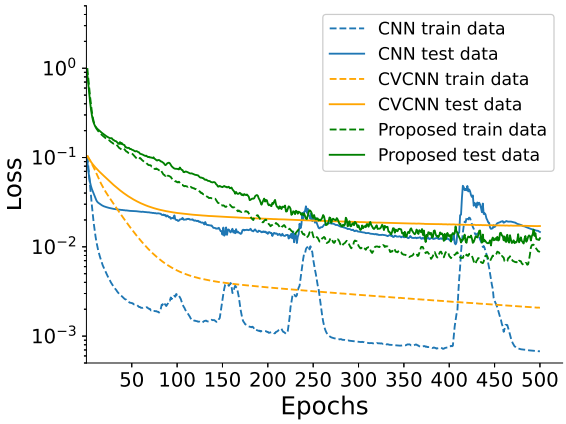
Fig. 2: (a) Classification accuracy for the first dataset with different data lengths from 0.25 s to 2.0 s with a step of 0.25 s. (b) Simulated ITRs across subject using different data lengths. The FBCCA method was quoted from [5]. The error bars indecate standard errors.

IV. Result

We compared classification accuracy and ITR. Fig. 2 shows the result of the first dataset. The proposed method showed the highest classification accuracy. The ITR of the proposed method also showed the highest value. It was demonstrated that the proposed method achieved higher accuracy compared to the FBCCA method from the reference study. Table. I shows the classification accuracy for each subject of the first dataset with 1s data length. In S1, S2, and S8, the proposed method showed higher accuracy than the conventional methods. Fig. 3 shows the learning curve of neural network models at one trial of S1. The proposed method shows that the difference in loss between training and testing is smaller than the other methods.

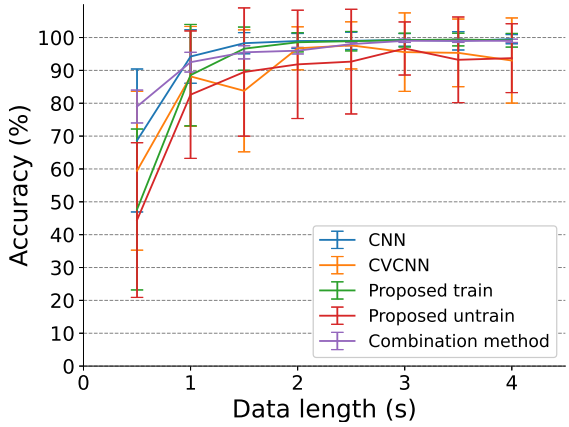


(a)

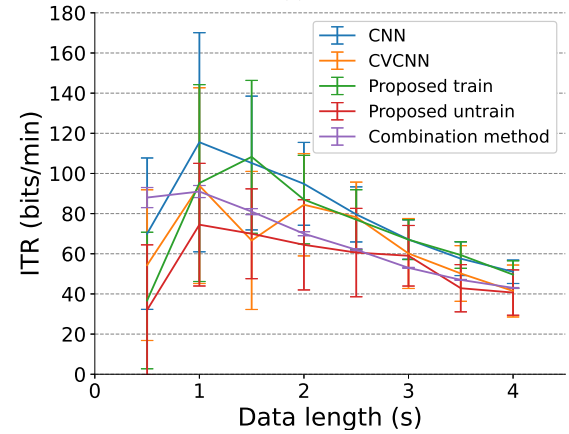


(b)

Fig. 3: (a) Learning curve accuracy and (b) learning curve loss for the first dataset with different data lengths from 0.25 s to 2.0 s with a step of 0.25 s. This is the result of one trial of subject1. "Proposed" model is "Proposed train" model.



(a)



(b)

Fig. 4: (a) Classification accuracy for the second dataset with different data lengths from 0.5 s to 4.0 s with a step of 0.5 s. (b) Simulated ITRs across subject using different data lengths. The combination method was quoted from [6]. The error bars indecate standard errors.

TABLE I: Classification accuracy (%) of the first dataset

	Proposed train	Proposed untrain	CNN	CVCNN
S1	95.00	76.67	87.50	85.83
S2	97.92	86.25	87.92	87.08
S3	99.17	95.00	98.75	98.33
S4	97.08	93.33	95.83	96.25
S5	95.00	87.50	96.67	88.75
S6	89.58	69.17	91.67	93.33
S7	73.75	57.92	75.83	37.92
S8	82.92	62.92	72.50	76.67
Avg.±STD	91.30±8.85	78.59±14.09	88.33±9.66	83.02±19.45

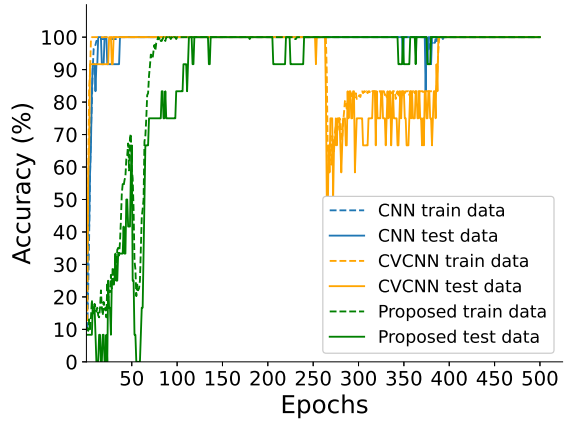
TABLE II: Classification accuracy (%) of the second dataset

	Proposed train	Proposed untrain	CNN	CVCNN
S1	79.44	65.00	95.56	92.78
S2	51.67	42.22	76.11	55.56
S3	89.44	76.67	100	100
S4	99.44	97.78	99.44	97.78
S5	100	98.33	100	96.67
S6	98.88	97.78	99.44	93.89
S7	99.44	95.56	98.89	97.22
S8	97.22	94.44	98.33	91.11
S9	92.78	92.78	95.56	93.33
S10	77.22	65.56	82.78	64.44
Avg.±STD	88.56±15.42	82.61±19.40	94.22±8.14	88.17±15.21

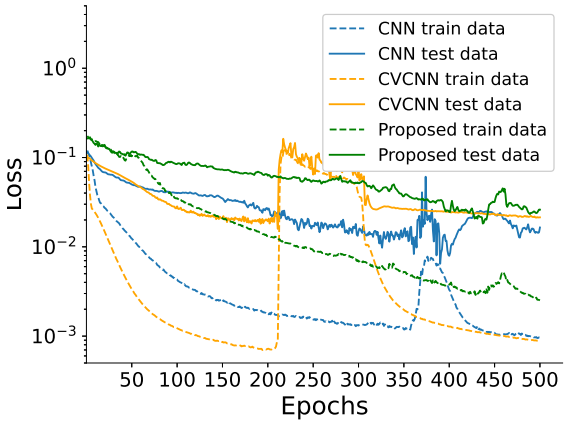
Fig. 4 shows the result of the second dataset. CNN has the highest classification accuracy. The proposed method was the second best classification accuracy and ITR. It was demonstrated that the proposed method achieved higher accuracy compared to the Combination method from the reference study [6]. Table. II shows the classification accuracy for each subject of the second dataset with 1s data length. In S7, the proposed method showed the highest accuracy than the conventional methods. Fig. 5 shows the learning curve of neural network models at one trial of S7. The proposed method shows that the difference in loss between training and testing is small and that the learning is stable.

V. Conclusion

The proposed method improved classification accuracy compared to the conventional method in the first dataset. ITR



(a)



(b)

Fig. 5: (a) Learning curve accuracy and (b) learning curve loss for the second dataset with different data lengths from 0.5 s to 4.0 s with a step of 0.5 s. This is the result of one trial of subject1. ”Proposed” model is ”Proposed train” model.

of the proposed method achieved 134.8 bits/min on average. The conventional method obtaines frequency information of the signal by the convolutional layer with respect to the time direction, but these models cannot obtain amplitude and phase information. The proposed method utilizes the initial phase information, thus that the proposed model showed higher classification accuracy for the dataset with a large number of frequencies used. In the proposed method, when the weights of the Hilbert transform filter were learned, an improvement in accuracy was observed. This is thought to be due to the optimization of the filter weights tailored to the data.

In the second dataset, there was one subject for whom the proposed method was the most accurate. The proposed method was more stable in learning the classification accuracy and loss than the conventional method. This indicates that overlearning is unlikely to occur and generalization performance is high. Because of the small number of frequencies used, we believe that the conventional method, which only performs frequency analysis, was sufficiently accurate.

The phase-modulated SSVEP data used in the experiment does not increase the number of stimuli based entirely on phase information while incorporating phase modulation. Future works will include considering phase-modulated SSVEP-BCI using stimuli with different initial phases at the same frequency. The proposed model may be even more accurate than the conventional method.

REFERENCES

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain-computer interfaces for communication and control,” *In Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 1–2, 2007.

[2] V. Patelia and M. S. Patel, “Brain computer interface: Appications and P300 speller overview,” *In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5, 2019.

[3] Z. Yue, X. Q. Shane, W. He, and Z. Zhiqiang, “Data analytics in steady-state visual evoked potential-based brain-computer interface: A review,” *IEEE Sensors Journal*, vol. 21, 2 2020.

[4] R. Aya and et al., “Braincomputer interface spellers: A review,” *Brain Sciences*, vol. 8, no. 4, p. 57, 2018.

[5] W. Yijun, C. Xiaogang, G. Xiaorong, and G. Shangkai, “A benchmark dataset for SSVEP-based brain-computer interfaces,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, 10 2017.

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

[7] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015.

[8] L. Sergey and S. Christian, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *International Conference on Machine Learning*, vol. 37, pp. 448–456, 2015.

[9] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, “Deep residual learning for image recognition,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[10] A. Igor and K. Zain, “Analysis of EEG using multilayer neural network with multi-valued neurons,” *In 2018 IEEE Second International Conference on Data Stream Mining Processing*, pp. 392–396, 2018.

[11] P. Musa, S. Baha, and D. Dursun, “A novel method for automated diagnosis of epilepsy using complex-valued classifiers,” *IEEE Journal of Biomedical and Health Informatics*, pp. 108–118, 2016.

[12] Z. Junming and W. Yan, "A new method for automatic sleep stage classification," vol. 11, pp. 1097–1110, 5 2017.

[13] Z. Zhimian, W. Haipeng, X. Feng, and J. Ya-Qiu, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 7177–7188, 12 2017.

[14] A. Ikeda and Y. Washizawa, "Spontaneous EEG classification using complex valued neural network," *International Conference on Neural Information Processing*, vol. 1142, pp. 495–503, 2019.

[15] T. Chiheb and et al., "Deep complex networks," *International Conference on Learning Representations*, 2018.

[16] K. Taehwan and A. Tulay, "Approximation by fully complex multilayer perceptrons," *Neural Computation*, vol. 15, pp. 1641–1666, 7 2003.