Convolutional Correlation Analysis for Enhancing the Performance of SSVEP-based Brain-computer Interface

Yao Li, Jiayi Xiang, and Thenkurussi Kesavadas

***Abstract*—Currently, most of the high-performance models for frequency recognition of steady-state visual evoked potentials (SSVEPs) are linear. However, SSVEPs collected from different channels can have non-linear relationship among each other. Linearly combining electroencephalogram (EEG) from multiple channels is not the most accurate solution in SSVEPs classification. To further improve the performance of SSVEP-based brain-computer interface (BCI), we propose a convolutional neural network-based non-linear model, i.e. convolutional correlation analysis (Conv-CA). Different from pure deep learning models, Conv-CA use convolutional neural networks (CNNs) at the top of a self-defined correlation layer. The CNNs function on how to transform multiple channel EEGs into a single EEG signal. The correlation layer calculates the correlation coefficients between the transformed single EEG signal and reference signals. The CNNs provide non-linear operations to combine EEGs in different channels and different time. And the correlation layer constrains the fitting space of the deep learning model. A comparison study between the proposed Conv-CA method and the task-related component analysis (TRCA) based methods is conducted. Both methods are validated on a 40-class SSVEP benchmark dataset recorded from 35 subjects. The study verifies that the Conv-CA method significantly outperforms the TRCA-based methods. Moreover, Conv-CA has good explainability since its inputs of the correlation layer can be analyzed for visualizing what the model learnt from the data. Conv-CA is a non-linear extension of spatial filters. Its CNN structures can be further explored and tuned for reaching a better performance. The structure of combining neural networks and unsupervised features has the potential to be applied to the classification of other signals.**

***Index Terms*—Brain-computer interface (BCI), convolutional correlation analysis (Conv-CA), deep learning, electroencephalogram (EEG), steady-state visual evoked potential (SSVEP).**

I. Introduction

The steady-state visual evoked potential (SSVEP) is one of the most popular electroencephalogram (EEG) signal for human-machine communication in the field of brain-computer interface (BCI). Because of the high information transfer rate (ITR), few user training, and ease of trigger, SSVEPs are broadly used in various of applications, such as medical [1,2], industrial [3,4], communication [5,6], smart home [7,8], gaming [9,10,11], and robot & vehicle control [12-16], etc. The performances and ITRs of SSVEP-based BCI applications largely depend on the classification accuracy of the SSVEP signal.

SSVEPs are brain response characterized by frequency pattern at stimulation frequency and its harmonic frequencies [17]. Intuitively, SSVEPs can be detected if transformed into frequency domain. One of the most powerful classification methods is the Power Spectra Density based Analysis (PSDA), which is based on the Discrete Fourier Transform. The PSDA method is designed to be use on the single channel EEG which results the method to be noise sensitive. Moreover, the DFT has low frequency resolution if time window of the EEG is short. To improve the PSDA method, Friman [18] linearly combined EEGs from different channels maximizing signal-noise ratios (SNRs). Meanwhile, Lin [19] came up a spatial filtering approach, i.e. the Canonical Correlation Analysis (CCA). CCA also linearly combines different channels to maximize SNRs, which is similar to Friman's "Maximum Contrast Combination", but it solves the frequency resolution drawback of the PSDA by calculating the correlations between EEG signal and artificial reference signals in arbitrary frequencies. Currently CCA is still the most popular SSVEP classification method and an important benchmark for new classification methods.

A concern for the CCA method is that its performance varies across different subjects, which means CCA may result low classification accuracy on some subjects. Many researches contribute to improve CCA. One of the most efficient directions is individual calibration data, i.e. constructing subject-specific model for each individual subject based on a training dataset. In CCA, the reference signals are artificial signals constructed with sine and cosine. Artificial signals lack of subject-specific information. Zhang et al. proposed the MwayCCA [19], L1-MCCA [20], and MsetCCA [21] based on standard CCA using subject-specific reference signals constructed from the training dataset for each subject. To further enhance the spatial filter on different harmonic frequency components, Chen et al. [22] built the filter bank CCA (FBCCA) which decompose SSVEPs into

Yao Li and Thenkurussi Kesavadas are with the Department of Industrial and Enterprise System Engineering, University of Illinois Urbana-Champaign, Champaign, Illinois, 61820.

Jiayi Xiang is with the Department of Electrical Computer Engineering, University of Illinois Urbana-Champaign, Champaign, Illinois, 61820.

multiple sub-band components under multiple pre-processing filters, then fusion the classifications from all sub-band. The training dataset was also utilized to optimize the weights of spatial filters instead of merely being used as reference signals. Wang et al. [23] extended the CCA structure with canonical correlations between training data, validation data, and reference signals. Nakanish et al. [5] introduced ensembled TRCA to optimize spatial filter weights by maximizing the reproducibility in cross-session training dataset. The ensemble TRCA achieved an averaged ITR of 325.33 bit/min in a 40-target cue-guided task. Xu et al. [24]. applied the ensemble TRCA on a hybrid BCI using concurrent P300 and SSVEP for 108 target characters and reached 172 bits/min on their on-line experiment. The CORCA-based method proposed by Zhang et al. [25] extracting maximally correlated signal components of EEG from multiple subjects was reported to have similar performance as TRCA on a benchmark dataset [26], which is around 220 bit/min on the 0.5 s time window length.

Recently, deep learning techniques have been successfully applied in various classification tasks in many domains. In EEG classification tasks, deep learning models can provide non-linear operations for the nonlinearities inevitably existed in the real signals [27, 28]. However, in the SSVEP frequency classification task, the performances of deep learning models haven't reach or not even close to simple spatial filters, such as TRCA or ensemble TRCA. Aznan et al. [29] introduced a CNN model in a 5-frequencies SSVEP classification task. They compared the CNN model with other SSVEP deep learning models, such as LSTM, SVM, and RNN, and demonstrated a better performance in their dataset. Similarly, Kwak et al. [30] developed another CNN based model in 5-target SSVEP experiment and obtained higher accuracy than the CCA method. However, these methods were neither investigated on a large number of targets nor validated on a standard benchmark dataset. Podmore et al. [31] proposed a deep CNN framework (PodNet) reached an ITR of 79.26 bit per minute in the same benchmark dataset [26] as CORCA. PodNet is the first deep learning model which tested on a benchmark dataset and compared with the FBCCA. Except the PodNet has much worse performance than the FBCCA. The main reason that the current deep learning models haven't succeed on SSVEP classification task is structure overfitting. Because SSVEPs have low signal-to-noise ratios (SNRs), most of the parameters in pure deep neural network frameworks are actually learning from noise. Even with large amount of training data provided, these deep neural networks are still not able to estimate the true model. Since noise is dominating SSVEP signals, classic neural networks, which works well in other domain, can't extract the signal from the training dataset.

Some deep learning models have been implemented and validated in the classification tasks for other EEG signals, such as P300, error-related negativity (ERN), movement-related cortical potential (MRCP), etc. EEGnet [32] with 3-layer CNNs was developed to classify various EEG signals including P300, ERN and MRCP. The first layer of EEGnet convoluted across sampling time points to calculate frequency features. The second layer convoluted across channels to combine EEG input channels pointwise. However, in SSVEP classification, we usually combine the channels first to obtain a better frequency pattern. MSFBCNN [33] was designed for MRCP classification and had a similar CNN structure as EEGnet. It used four convolutional kernels in different sizes in the first layer to extract frequency patterns in multiple scales. SCCNet [34] applied 2-layer CNNs with 2-dimentional convolutional kernels in both layers for MRCP classification. But both kernels contained only one filter which has limited ability for sufficient feature extraction. Moreover, in SSVEP classification, the training dataset average has been verified to be a better reference signal than the sine-cosine artificial reference signal as it contains subject-specific information. This information is difficult to be extracted in deep learning models given limited training dataset.

Thus, we proposed a new deep learning model for SSVEP classification, called convolution correlation analysis (Conv-CA). The Conv-CA combines CNN structure and traditional correlation analysis. It has two CNN branches named signal-CNN and reference-CNN. The signal-CNN provides convolutional operation for the multi-channel EEG signals in every tiny time windows. It transforms EEGs collected from multi-channels into a single signal. Similarly, reference-CNN convolutionally combines reference signals from multiple channels and outputs a one-channel reference signal for each frequency. A correlation layer connecting the signal-CNN and the reference-CNN calculates the correlations between the EEG signal and the reference signal. The CNN structure extends the spatial filter to a non-linear model. Meanwhile, the correlation layer constrains the fitting space of the deep learning model. The Conv-CA is designed to classify SSVEPs using only single subject data, so we compared our model with the ensemble TRCA method [5] which is one of the state-of-art SSVEP classification model for individual subject.

The rest of the paper is organized as follows. Section II introduces a benchmark dataset, the TRCA-based methods, our proposed Conv-CA model, the evaluation metrics, and a frequency pattern extraction method. Section III demonstrates experiment results of the TRCA-based methods and the Conv-CA model on the benchmark dataset. It also contains analyses for the inputs of the correlation layer, the performance of Conv-CA on low-accuracy subjects, smaller training dataset, and hyperparameter further tuning. Section IV explains the significance of the proposed Conv-CA model. In Section V, we discussed some future research directions. Conclusion is given in the last section. Our implementation is available at GitHub[1]. As validating our proposed Conv-CA takes a lot of computation resource, to ease future research referring our work, we also uploaded the classification accuracies for every subject in the dataset.

[1] https://github.com/yaoli90/Conv-CA

II. Methods and Materials

A. Benchmark Dataset

Our model Conv-CA and the TRCA [5] are validated on a SSVEP benchmark dataset [26]. The dataset was collected through a cue-guided target selecting task in an offline BCI experiment. EEGs from thirty-five healthy subjects (seventeen females, eighteen males, mean age: twenty-two years) were collected. The experiment included six trials for each subject. Each trial contains 40 simulation tests corresponding to 40 different characters in random order. In each test, subjects were instructed to stare at a target character without eye blinks during the stimulation duration. Each test started with a 0.5-second target cue. Before cue ends, subjects were asked to shift their gaze to the target character. Following the cue, all stimuli flicked on the screen concurrently for 5 seconds. After the 5-seconds simulation, the screen was blanked for 0.5 second before the start of next test.

The 40 target characters of the BCI were arranged in a 5 × 8 matrix on a monitor. The characters were coded in joint frequency and phase modulation (JFPM) approach. The frequencies of the 40 targets were from 8 Hz to 15.8 Hz with a 0.2 Hz offset between two targets. The phase offset between two characters was 0.5 π. EEG data were recorded through a Synamps2 EEG system (Neuroscan, Inc.) under a sampling rate of 1000 Hz. Sixty-four electrodes were placed on the standard positions according to an extended 10-20 system. A reference electrode was placed at the vertex (Cz). During the experiment, electrode impedances were kept below 10 kΩ.

A 50 Hz notch filter was applied to recorded EEGs to remove the power-line noise. All the epochs were down sampled to 250 Hz subsequently.

B. Data Preprocessing

All EEG data were filtered with a Chebyshev Type I filter with cutoff frequencies from 6 Hz to 90 Hz and stopband corner frequencies from 4 Hz to 100 Hz. Because the EEG recording in each test contains 0.5 second pre-stimulus and 0.5 second post-stimulus, we only used the data from 0.5 second to 5.5 seconds for frequency recognition.

C. TRCA-Based Method

TRCA is a method that learns the weights of spatial filters from the training dataset of an individual subject. The weights are calculated by maximizing the similarity of EEGs among all trials in the training dataset. Denote the EEG signals as $X \in \mathbb{R}^{N_s \times N_c \times N_f \times N_t}$, where N_s, N_c, N_f, and N_t are the number of sampling points, the number of channels, the number of target SSVEP frequencies, and the number of trials, respectively. For a frequency target $n \in [1, N_f]$, denote the weight of the corresponding spatial filter as $\boldsymbol{w}_n \in \mathbb{R}^{N_c \times 1}$. The covariance among different trials is

$$\begin{aligned}&\sum_{i,j=1,i\neq j}^{N_t} Cov\left(\boldsymbol{X}^{(*,*,n,i)}\boldsymbol{w}_n, \boldsymbol{X}^{(*,*,n,j)}\boldsymbol{w}_n\right)\\ &= \sum_{i,j=1,i\neq j}^{N_t} \boldsymbol{w}_n^T \boldsymbol{X}^{(*,*,n,i)^T}\boldsymbol{X}^{(*,*,n,j)}\boldsymbol{w}_n\\ &= \boldsymbol{w}_n^T \sum_{i,j=1,i\neq j}^{N_t} \boldsymbol{X}^{(*,*,n,i)^T}\boldsymbol{X}^{(*,*,n,j)}\,\boldsymbol{w}_n\\ &= \boldsymbol{w}_n^T \boldsymbol{S} \boldsymbol{w}_n\end{aligned} \quad (1)$$

Here, the $\boldsymbol{X}^{(*,*,n,i)}$ and $\boldsymbol{X}^{(*,*,n,j)}$ denote the n-th frequency in the i-th and j-th trials of $\boldsymbol{X}$.

The covariance among all the trials is

$$\begin{aligned}&\sum_{i,j=1}^{N_t} Cov\left(\boldsymbol{X}^{(*,*,n,i)}\boldsymbol{w}_n, \boldsymbol{X}^{(*,*,n,j)}\boldsymbol{w}_n\right)\\ &= \boldsymbol{w}_n^T \boldsymbol{Q} \boldsymbol{w}_n\end{aligned} \quad (2)$$

Note that the equation (2) has different form as represented in [5], but they are mathematically equivalent.

The weight $\boldsymbol{w}_n$ of the spatial filter for the target n can be calculated by maximizing the covariance among different trials divided by the covariance among all trials, i.e.

$$\boldsymbol{w}_n^* = \underset{\boldsymbol{w}_n}{\operatorname{argmax}} \frac{\boldsymbol{w}_n^T \boldsymbol{S} \boldsymbol{w}_n}{\boldsymbol{w}_n^T \boldsymbol{Q} \boldsymbol{w}_n} \quad (3)$$

The optimal weight $\boldsymbol{w}_n^*$ can be calculated as the eigenvector of matrix $\boldsymbol{Q}^{-1}\boldsymbol{S}$ corresponding to the largest eigenvalue. We denote the above steps for obtain optimal $\boldsymbol{w}_n^*$ as a function

$$\boldsymbol{w}_n^* = \gamma(\boldsymbol{X}, n) \quad (4)$$

For an individual subject with a training dataset $\boldsymbol{X}_{train} \in \mathbb{R}^{N_s \times N_c \times N_f \times N_t}$ and test data $\boldsymbol{X}_{test} \in \mathbb{R}^{N_s \times N_c \times N_f}$, the reference signal $\boldsymbol{Y} \in \mathbb{R}^{N_s \times N_c \times N_f}$ is the trial-averaged training data, i.e.

$$\boldsymbol{Y}^{(*,*,n)} = \frac{1}{N_t}\sum_{i=1}^{N_t} \boldsymbol{X}_{train}^{(*,*,n,i)}, \quad n \in [1, N_f] \quad (5)$$

The correlation coefficients between the test data $\boldsymbol{X}_{test}$ and its reference signal $\boldsymbol{Y}$ are

$$\lambda_n = \rho\left(\boldsymbol{X}_{test}^{(*,*,n)}\boldsymbol{w}_n^*, \boldsymbol{Y}^{(*,*,n)}\boldsymbol{w}_n^*\right), \quad n \in [1, N_f] \quad (6)$$

where $\boldsymbol{w}_n^* = \gamma(\boldsymbol{X}_{train}, n)$ and $\rho(.,.)$ indicates the one-dimensional correlation analysis. Then, the frequency of the test data $\boldsymbol{X}_{test}$ can be recognized by

$$f_n, \quad n = \operatorname{argmax} \lambda_n, \quad n \in [1, N_f] \quad (7)$$

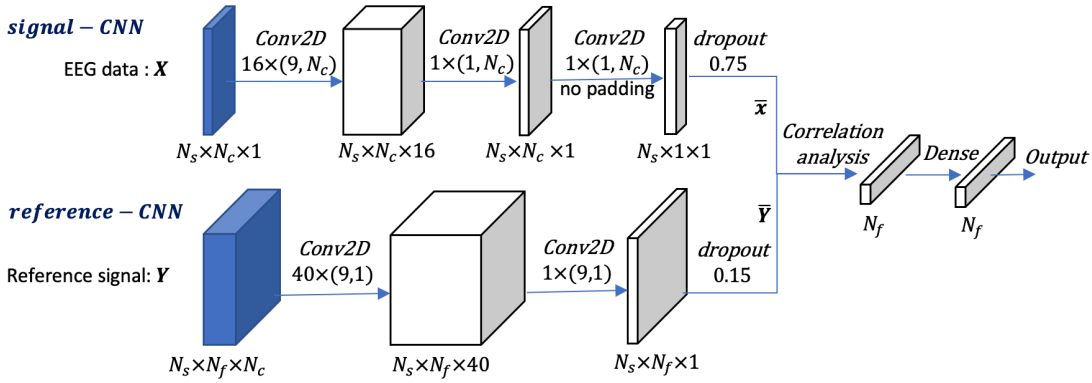

Fig. 1. Structure of the proposed Conv-CA SSVEP classification model. The cuboids represent the shape of output tensors. The CNN layers use zero paddings except the second layer of the signal-CNN which has no padding. All the CNNs use linear activation functions. The activation function of the dense layer is softmax. See detailed architecture in Table 1.

Table 1. Conv-CA architecture, where N_s = number of time sampling points, N_c = number of channels, N_f = number of target frequencies (number of classes). In our implementation, $N_s = 50, 100, 150, 200$ and 250, corresponding to 0.2 s, 0.4 s, 0.6 s, 0.8 s and 1.0 s of EEG segmentations. $N_c = 9$, channels are Pz, PO5, PO3, POz, PO4, PO6, O1, Oz, and O2. $N_f = 40$ corresponding to 40 characters flashing in 40 different frequencies. The Corr layer is a self-defined layer. It takes the outputs of the signal-CNN and the reference-CNN as inputs and calculates their correlation coefficients. The Corr layer doesn't contain any trainable parameter and it involves in the process of backpropagation when the model trains. See Section II. F for more details.

	Layer	Kernel size	# Filters	# Parameters	Output	Activation	Padding
signal-CNN	Input				$(N_s \times N_c \times 1)$		
	Conv2D	$(9, N_c)$	16	$(9 \times N_c + 1) \times 16$	$(N_s \times N_c \times 16)$	linear	same
	Conv2D	$(1, N_c)$	1	$N_c \times 16 + 1$	$(N_s \times N_c \times 1)$	linear	same
	Conv2D	$(1, N_c)$	1	$N_c + 1$	$(N_s \times 1 \times 1)$	linear	valid
	Dropout		$p = 0.75$				
	Flatten				(N_s)		
reference-CNN	Input				$(N_s \times N_f \times N_c)$		
	Conv2D	$(9, 1)$	40	$(9 \times N_c + 1) \times 40$	$(N_s \times N_f \times 40)$	linear	same
	Conv2D	$(9, 1)$	1	$9 \times 40 + 1$	$(N_s \times N_f \times 1)$	linear	same
	Dropout		$p = 0.15$				
Correlation analysis	Input				$(N_s) + (N_s \times N_f \times 1)$		
	Corr (self-defined)			0	(N_f)		
Classifier	Dense	(N_f)		$N_f \times N_f$	(N_f)	softmax	

D. Ensemble TRCA-based Method

Experiment results shows that integrating all spatial filters of all frequency targets could further improve the performance of TRCA. With N_f spatial filters, a concatenated matrix $\boldsymbol{W}^* \in \mathbb{R}^{N_c \times N_f}$ can be obtained. Then the equation (6) can be replaced by

$$\lambda_n = \rho\left(\boldsymbol{X}_{test}^{(*,*,n)}\boldsymbol{W}^*, \boldsymbol{Y}^{(*,*,n)}\boldsymbol{W}^*\right), \quad n \in [1, N_f] \qquad (8)$$

where $\rho(.,.)$ indicates the two-dimensional correlation analysis instead.

E. TRCA with Filter Bank

Filter bank technique have been verified to improve the performance of TRCA-based method in [5]. The filter bank analysis decomposes SSVEP signals into sub-band components using infinite impulse response (IIR) filters, such that the harmonic components of SSVEP signals can be embedded independently. According to the study [22], the lower and upper cut-off frequencies of the *m-th* sub-band were set to $m \times 8$ Hz and 90 Hz, where $m \in [1, N_b]$ and N_b is the number of sub-bands. By applying the *m-th* Chebyshev Type I filters to the training dataset $\boldsymbol{X}_{train}$ and test data $\boldsymbol{X}_{test}$, a sub-band correlation for the frequency target n can be obtained as λ_n^m from equation (6) or equation (8). With all sub-bands, a set of correlations $\boldsymbol{\beta}_n = \left(\lambda_n^1, \lambda_n^2, \ldots, \lambda_n^{N_b}\right)^T$ for the target n can be linear combined by

$$\lambda_n = \boldsymbol{\Phi}^T \boldsymbol{\beta}_n \qquad (9)$$

where $\boldsymbol{\Phi}$ is defined as

$$\boldsymbol{\Phi} = (1^{-a} + b \quad 2^{-a} + b \quad \ldots \quad N_b{}^{-a} + b)^T \qquad (10)$$

Here a and b are empirically set to 1.25 and 0.25 respectively.

F. Conv-CA Design

The Conv-CA is designed to classify multi-target SSVEP signals with a structure combining CNN and correlation

analysis. The detailed structure is demonstrated in Fig. 1 and Table 1. The Conv-CA takes two inputs, i.e. the EEG data $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c \times 1}$ and the reference data $\boldsymbol{Y} \in \mathbb{R}^{N_s \times N_c \times N_f}$. In our implementation, we use the 9 channels as same as the channels used in the TRCA method, which are Pz, PO5, PO3, POz, PO4, PO6, O1, Oz, and O2. To simplify the implementation, we rearranged the order of dimensions of $\boldsymbol{Y}$ from $\mathbb{R}^{N_s \times N_c \times N_f}$ to $\mathbb{R}^{N_s \times N_f \times N_c}$.

We apply a 3-layer CNN to the EEG data $\boldsymbol{X}$ as

$$\overline{\boldsymbol{x}} = f(\boldsymbol{X}), \quad \overline{\boldsymbol{x}} \in \mathbb{R}^{N_s} \tag{11}$$

Here the 3-layer CNN $f(.)$ is named as signal-CNN and shown in the top branch in Fig. 1. The first layer of signal-CNN has 16 filters of $9 \times N_c$ kernels. It convolutes EEGs in all input channels ($N_c = 9$) in a short local time period (9 sampling points or 36 ms). The two-dimensional kernels in this layer extends the pointwise channel combination methods, which are used in CCA, TRCA, EEGnet, etc., to a short local time interval channel combination. The second layer combines the 16 filters in the first layer into one. It uses $1 \times N_c$ kernels to weight EEGs from different channels. This layer is designed to join the frequency pattens obtained from the first layer. The third layer applies a $1 \times N_c$ kernel with no padding (The first and second layers use zero paddings to keep the output to have same dimension as the input.) to transform the data $\boldsymbol{X} \in \mathbb{R}^{N_s \times N_c \times 1}$ into a one-dimension signal $\overline{\boldsymbol{x}} \in \mathbb{R}^{N_s}$. At the end of signal-CNN, we apply a dropout with dropping rate 75% to $\overline{\boldsymbol{x}}$ for regularization.

Another 2-layers CNN is applied to the reference signal $\boldsymbol{Y}$ as

$$\overline{\boldsymbol{Y}} = g(\boldsymbol{Y}), \quad \overline{\boldsymbol{Y}} \in \mathbb{R}^{N_s \times N_f \times 1} \tag{12}$$

where $g(.)$ is named as reference-CNN. The first layer contains 40 filters of 9×1 kernels. It provides convolutional weights for each class in all channel EEGs across a short time period. The second layer uses 9×1 kernels to integrate the filters. Similar as the signal-CNN, a dropout layer with dropping rate 15% is added at the end of the reference-CNN. Note that, in the reference-CNN, the convolutional operations are applied on each class independently.

The Corr layer in the correlation analysis takes $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{Y}}$ as inputs and calculates the correlation coefficients of $\overline{\boldsymbol{x}}$ and each $\overline{\boldsymbol{Y}}^{(.,n)}$ for all $n \in [1, N_f]$. By denoting $\overline{\boldsymbol{y}}_n = \overline{\boldsymbol{Y}}^{(.,n)}$, the correlation coefficients in the correlation analysis layer are

$$\sigma(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_n) = \frac{\sqrt{(\overline{\boldsymbol{x}} \circ \overline{\boldsymbol{x}})^T (\overline{\boldsymbol{y}}_n \circ \overline{\boldsymbol{y}}_n)}}{\sqrt{\overline{\boldsymbol{x}}^T \overline{\boldsymbol{x}}} \sqrt{\overline{\boldsymbol{y}}_n^T \overline{\boldsymbol{y}}_n}}, \quad n \in [1, N_f] \tag{13}$$

where $\circ$ is the Hadamard product (i.e. element-wise product). Then the output of the Corr layer is

$$\begin{aligned} \boldsymbol{z} &= h(\overline{\boldsymbol{x}}, \overline{\boldsymbol{Y}}) \\ &= \left[\sigma(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_1) \quad \sigma(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_2) \quad \dots \quad \sigma\left(\overline{\boldsymbol{x}}, \overline{\boldsymbol{y}}_{N_f}\right)\right] \in \mathbb{R}^{N_f} \end{aligned} \tag{14}$$

We use a dense layer with N_f units and softmax activation function as the final layer for classification.

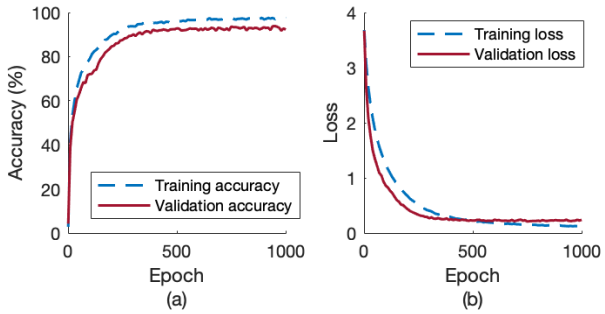

Fig. 2. (a) The training accuracy, the validation accuracy, (b) the training loss, and the validation loss of the first 5 trials EEG of subject 35 on 1.0 s time window over 1,000 training epochs.

Because we build a correlation analysis layer at the end of CNNs, the CNNs do not have to be active by non-linear activation functions. In our implementation, we use linear activation functions in all CNNs. The hyperparameters in Conv-CA are short local time period = 9, number of filters in signal-CNN = 16, dropout rate of signal-CNN = 0.75, number of filters in reference-CNN = 40, and dropout rate of reference-CNN = 0.15. These parameters are the tuning result based on the data of the subject 1. And the tuned model is directly applied to the other 34 subjects without any further tuning. Since the EEG data is noisier than the reference signal, the signal-CNN has much larger dropout rate than the reference-CNN even though the signal-CNN has less trainable parameters.

The Conv-CA is implemented in python-Keras with tensorflow backend. We use categorical cross-entropy as loss function. The optimization is solved with Adam algorithm (learning rate (8e-4), beta1 (0.9), beta2 (0.999)) with a batch size 32. The model runs on the Google Cloud platform.

G. Performance Evaluation

Accuracy of Classification and ITR are used to evaluate the performance of all models. While calculating ITRs, the 0.5 second gaze-shifting time from the original off-line experiment is added. We evaluate all the methods in various data lengths (0.2 s to 1 s, with a step of 0.2 s). As the dataset has 0.5 s in the beginning of each trial for target cue and 0.5 s blank at the end of each trial, we extracted the data from 0.5 s to 5.5 s, which is the simulation period of each 6 s trial. For the input of the signal-CNN branch, we applied a sliding window with the step of $0.15 \times data\ length$ to generate each data sample. For example, when $N_s = 100$ (0.4 s), there are $0.15 \times N_s = 15$ time points of non-overlapping for each two adjacent data samples. For the input of the reference-CNN, we first averaged the EEG recordings in all the training dataset, then applied the same sliding window as above. Since the EEG data input and the reference signal are generated using the same sliding window, they pair with each other and their phases are aligned to each other. In the testing stage, we pair each EEG data sample with the reference signal generated in the training stage as inputs. The sampling method is the same as in [5].

TRCA-based methods are used to compare with our proposed Conv-CA model. We use leave-one-out cross validation to evaluate the methods. Specifically, one of the 6

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2020.3038718, IEEE Transactions on Neural Systems and Rehabilitation Engineering

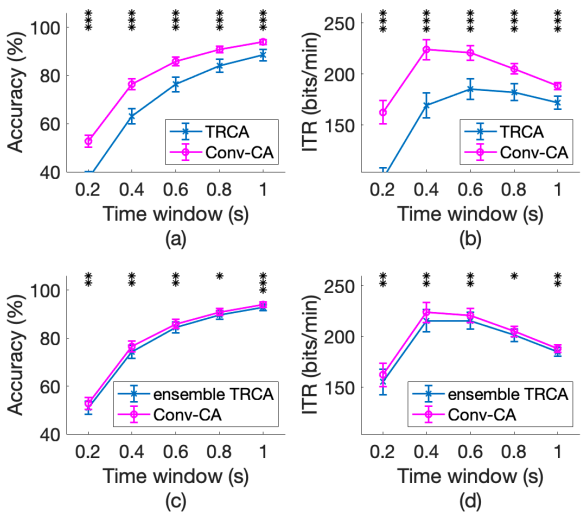

Fig. 3. The averaged accuracies (a, c) and ITRs (b, d) across all subjects by TRCA-based methods and Conv-CA model in various time windows. The asterisks in subfigures indicate significant difference between two methods by paired t-tests ($*p < 0.05$, $**\ p < 0.01$, $***\ p < 0.001$). The error bars indicate standard errors.

trials of the EEG data are used as test data and the other 5 trials are used as training dataset. The reference signal is calculated using the entire training dataset and being used in both training and testing. This process is repeated for 6 times so that every trial is tested. Paired t-test are performed to verify the statistical differences of the Conv-CA model and TRCA-based methods.

The Conv-CA model is trained with a Tesla K80 GPU. When the time window $N_s = 50$ (0.2 s), each epoch takes 11 s. The model converges within 100 epochs. To avoid gradient exploding, a gradient clipping at 5.0 is added in the Adam solver. For $N_s = 100$ (0.4 s), $N_s = 150$ (0.6 s) and 200 (0.8 s), the model converges within 500 epochs while each epoch takes 9 s, 8 s, and 7 s respectively. As for $N_s = 250$ (1.0 s), the model converges in 1,000 epochs. Each epoch takes 7 s. We demonstrate an example of the epoch selection in Fig. 2. Fig. 2. shows the training loss, validation loss, training accuracy, and validation accuracy while training Conv-CA on the first 5 trials EEG data of subject 35 with $N_s = 250$. In Fig. 2. (a), the training accuracy and the validation accuracy increase rapidly in the first 500 epochs. Then they slowly grow and converge in the rest 500 epochs. In Fig. 2. (b), the training loss decreases over the entire 1,000 epochs. However, the validation loss almost stops decreasing in the last 500 epochs. Fig. 2. shows that the model converges within 1,000 training epochs. Similar training loss and validation loss ensure the model have similar performance on both the training dataset and the test data.

The Conv-CA is compared with two TRCA-based methods, i.e. basic TRCA and ensemble TRCA. According to [5], the basic TRCA and the ensemble TRCA achieved best performance with 5 sub-bands. Thus, we applied filter bank technique on both methods with 5 sub-bands.

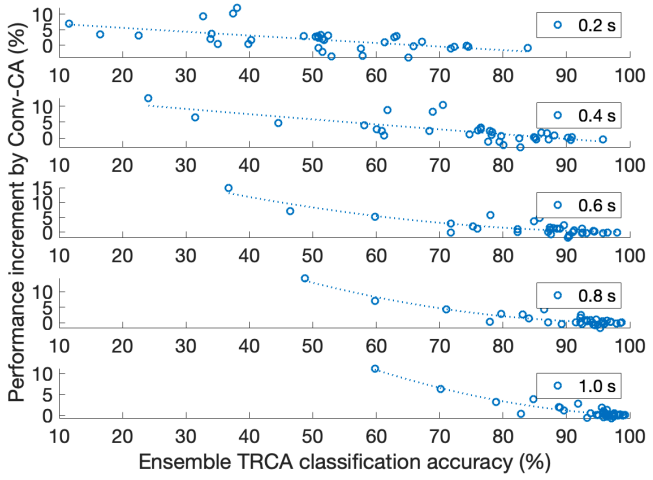

Fig. 4. The Conv-CA performance increments from ensemble TRCA vs the classification accuracies of ensemble TRCA method in various time windows. The dash lines are 2-nd order polynomial curve fittings of the increments to visualize the trends.

H. Frequency Pattern Extraction

One of the important advantages of the Conv-CA model is its good explainability. We can visualize what the model extracted from the EEG data and the reference signal by analyzing the intermediate tensors, i.e. $\bar{\boldsymbol{x}} \in \mathbb{R}^{N_s}$ and $\bar{\boldsymbol{Y}} \in \mathbb{R}^{N_s \times N_f \times 1}$. The most important information for SSVEPs is the frequency pattern. To obtain the high-resolution frequency pattern for a one-dimentional signal $\hat{\boldsymbol{x}}$ on every target frequency $f_n, n \in [1, N_f]$, we propose to use the correlation coefficients of $\hat{\boldsymbol{x}}$ and a series of artificial sine-cosine signals $\hat{\boldsymbol{Y}}_n$. This implementation is similar as the CCA method in [19]. We construct $\hat{\boldsymbol{Y}}_n \in \mathbb{R}^{N_s \times 2H}$ as

$$\hat{\boldsymbol{Y}}_n = \begin{bmatrix} \sin 2\pi f_n t \\ \cos 2\pi f_n t \\ \vdots \\ \sin 2\pi H f_n t \\ \cos 2\pi H f_n t \end{bmatrix}^T, \quad t = \left[\frac{0}{F_s}, \cdots, \frac{N_s - 1}{F_s}\right] \tag{15}$$

Here, H is the number of harmonics. We choose $H = 5$, because the cutoff frequencies are 6 Hz to 90 Hz in the preprocessing. $F_s = 250$ is the sampling frequency after down sampling. The frequency pattern for $\hat{\boldsymbol{x}}$ is extracted as

$$\boldsymbol{p}(\hat{\boldsymbol{x}}) = \left[CCA\left(\hat{\boldsymbol{x}}, \hat{\boldsymbol{Y}}_1\right) \quad \cdots \quad CCA\left(\hat{\boldsymbol{x}}, \hat{\boldsymbol{Y}}_{N_f}\right)\right] \in \mathbb{R}^{N_f} \tag{16}$$

where

$$CCA(\boldsymbol{x}, \boldsymbol{y}) = \max_{\boldsymbol{w},\boldsymbol{v}} \frac{E[\boldsymbol{w}^T \boldsymbol{x} \boldsymbol{y}^T \boldsymbol{v}]}{\sqrt{E[\boldsymbol{w}^T \boldsymbol{x} \boldsymbol{x}^T \boldsymbol{w}] E[\boldsymbol{v}^T \boldsymbol{y} \boldsymbol{y}^T \boldsymbol{v}]}} \tag{17}$$

Since $CCA\left(\hat{\boldsymbol{x}}, \hat{\boldsymbol{Y}}_n\right)$ is mathematical proportional to the square root of the sum of energy spectral density of $\hat{\boldsymbol{x}}$ on f_n and its harmonics. Intuitively, the frequency pattern $\boldsymbol{p}(\hat{\boldsymbol{x}})$ represents the energy of $\hat{\boldsymbol{x}}$ on each target frequency and its harmonics.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSRE.2020.3038718, IEEE Transactions on Neural Systems and Rehabilitation Engineering

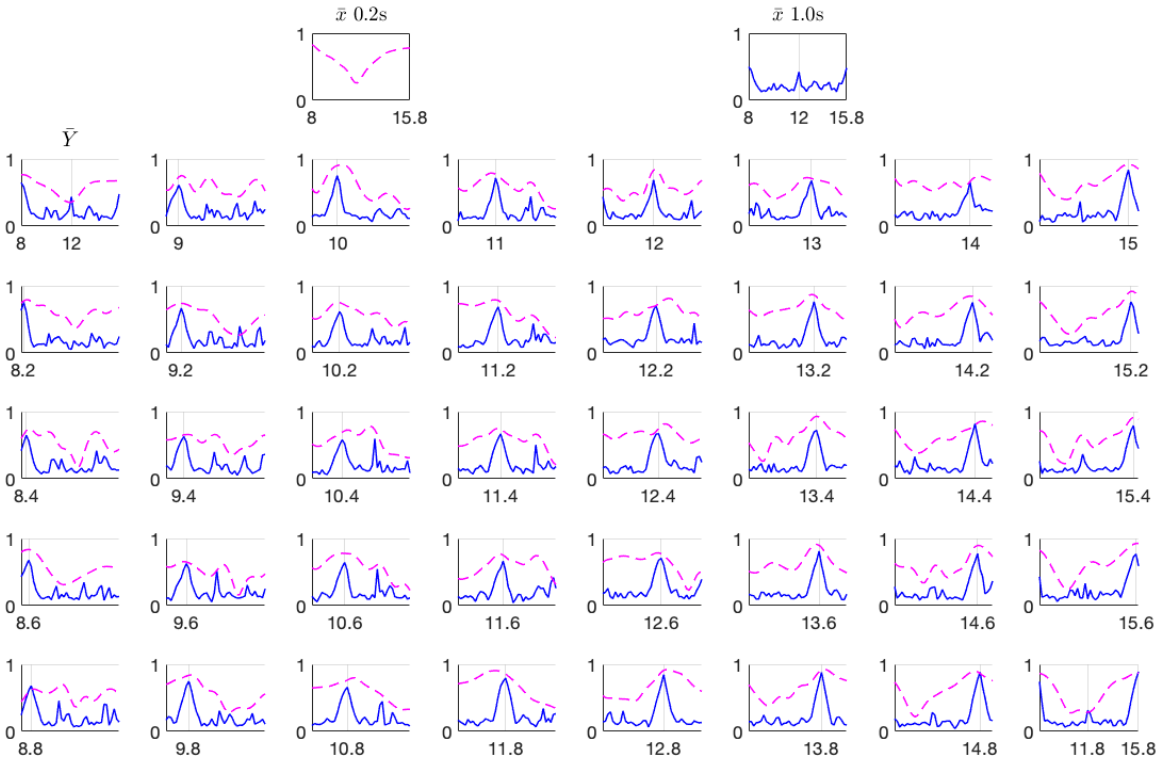

Fig. 5. The frequency patterns for the inputs ($\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{Y}}$) of the correlation analysis layer in Conv-CA. The pink dish line represents the first sample, i.e. a random sample from the testing process of subject 31 with length $N_s = 50$ (0.2 s) in 8 Hz. The bule solid line represents the second sample, which is also a random sample from the testing process of subject 31 in 8 Hz, but with length $N_s = 250$ (1.0 s). The top two subplots are frequency patterns for $\bar{\boldsymbol{x}}$ and the bottom 40 subplots are frequency patterns for $\bar{\boldsymbol{Y}}$. Each subplot in the bottom corresponding to one class in $\bar{\boldsymbol{Y}}$. The x tick is the target frequencies of the class. In the first and the last subplot, we added one extra tick at 12 Hz and 11.8 Hz to help illustrate our analysis. The first subplot is the class 8 Hz and the last plot is the class 15.8 Hz.

This frequency pattern extraction method is applied in the analysis in III.C.

III. Results

A. Classification Accuracy

We investigated the performance of our proposed Conv-CA model by comparing with TRCA-based methods. Because, as mentioned above, filter bank technique (5 sub-bands) can substantially improve the classification accuracy of TRCA-based methods, we only explored the basic TRCA method under 5 sub-bands and the ensemble TRCA method under 5 sub-bands. For simplification, we will mention them as TRCA and ensemble TRCA in below. Fig. 3 shows the classification accuracy and ITR comparisons across all subjects in various time windows. By fig. 3 (a) and (b), our proposed Conv-CA model significantly outperform ($p < 0.001$) the basic TRCA method in both classification accuracies and ITRs across 5 tested time windows from 0.2 s to 1.0 s. Fig. 3 (c) and (d) demonstrate that the Conv-CA model is significantly ($p < 0.05$) better than the ensemble TRCA method in both accuracies and ITRs across all time windows. The Conv-CA model achieves the highest ITR 226.19 bits/min on the 0.4 s time window, which is higher than the highest ITR 216.34 bits/min of the ensemble TRCA achieved on the same time window. The highest ITR of the Conv-CA is significantly ($p < 0.001$) higher than the highest ITR of the ensemble TRCA method.

B. Performance on Lower Accuracy Subjects

We also investigated how our proposed Conv-CA model performs on those subjects which have relative lower classification accuracies under TRCA-based methods. In Fig. 4, we plotted the absolute performance increment of Conv-CA compare to the ensemble TRCA vs the classification accuracy of the ensemble TRCA. For subjects whose SSVEP were not well-classified by the ensemble TRCA, the Conv-CA is more likely to have better performance than the ensemble TRCA. Especially in the 0.8 s time window, the classification accuracy of subject 33 increased from 43.57% to 61.71%. And the accuracy of subject 19 increased from 58.93% to 71.57%. This result shows that the Conv-CA extracts more useful information on the subjects whose SSVEP are difficult for ensemble TRCA to classify.

C. Correlation Analysis Inputs

To visualize what Conv-CA learnt from the training dataset. we extracted the frequency patterns of the inputs (i.e. $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{Y}}$) of the Corr layer using equation (16). In Fig. 5, we plotted the frequency patterns of $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{Y}}$ of two random samples from the testing process of subject 31. The first sample has $N_s = 50$ and the second one has $N_s = 250$. Both of them are in the class of

8 Hz. We chose the subject 31 because of its good classification accuracy. Its classification accuracies under the Conv-CA method are 83.13% for $N_s = 50$ and 99.06% for $N_s = 250$.

The top two subplots of Fig. 5 are $\boldsymbol{p}(\bar{\boldsymbol{x}})$ of the two samples. The bottom 40 subplots are $\boldsymbol{p}\left(\bar{\boldsymbol{Y}}^{(*,i,*)}\right), i \in [1, N_f]$, where the index i corresponding to the i-th target frequency (or class) in $\bar{\boldsymbol{Y}}$. In Fig. 5, for the second sample ($N_s = 250$), every $\bar{\boldsymbol{Y}}^{(*,i,*)}$ has a highest peak on its corresponding frequency. As $\bar{\boldsymbol{Y}}$ is the output of the reference-CNN, it implies that the reference-CNN learnt the information of the target frequency from the reference signal. The first sample ($N_s = 50$) has a much smoother frequency pattern than the second sample. For about half of the classes, it has a highest peak at the target frequencies. It implies that the model also learnt frequency patterns from other frequencies rather than the target frequency alone.

The first subplot on the top is the frequency pattern of $\bar{\boldsymbol{x}}$ of the first sample. The curve is similar as the 8 Hz class of $\bar{\boldsymbol{Y}}$ (the first subplot of the bottom 40 subplots) rather than other classes (other subplots of the bottom 40 subplots). As the curve represents the signal energy in all target frequencies, the similarity of the curves implies that the EEG signal has similar frequency patterns as the reference signal in all target frequencies. The $\bar{\boldsymbol{x}}$ plot of the second sample is similar with both $\bar{\boldsymbol{Y}}$ at 8 Hz and 15.8 Hz (the last subplot). However, it has a peak on 12 Hz which is the same as the $\bar{\boldsymbol{Y}}$ in 8 Hz, while $\bar{\boldsymbol{Y}}$ in 15.8 Hz has a peak on 11.8 Hz. As we have similar findings in other subjects and samples, the Conv-CA learnt frequency information from both the target frequency and other frequencies. We can further infer that the other frequencies are helpful in SSVEP classification.

D. Training Trial Reduction Analysis

In the data preparation, we used a step of $0.15 \times data\ length$ to generate training data from the recording trials. For $N_s = 50$, we got 30,000 training trials for each subject. As N_s increased to 250, there were 5,200 training trials for each subject. Because the number of training trials directly relates to the number of recordings trials in the experiments, reducing training trials plays an important role in real application. We analyzed classification accuracies when applying a sub-training dataset for subject 1, 2, 3, 4 and 5. In Fig. 6, we plotted the averaged classification accuracy of the 5 subjects with $N_s = 50$ (0.2 s), $N_s = 100$ (0.4 s), $N_s = 150$ (0.6 s), $N_s = 200$ (0.8 s), and $N_s = 250$ (1.0 s) using 2, 3, 4, and 5 recording trials for training. As we decreased the number of training trials, the performance of the Conv-CA model decreased in all 5 tested time windows. This performance decrease exists in all the 5 subjects although in Fig. 6 we only plotted their average. If the Conv-CA is trained with 4 recording trials, its performance decrease 198 bp, 163 bp, 162 bp, 157 bp and 154 bp with respect to 0.2 s, 0.4 s, 0.6 s, 0.8 s, and 1.0 s time window on these 5 subjects.

E. Hyperparameter Tuning

As mentioned in II.F, the hyperparameters of Conv-CA were tuned based on the data of the subject 1. The short local time period = 9, number of filters in signal-CNN = 16, dropout rate of signal-CNN = 0.75, number of filters in reference-CNN = 40, and dropout rate of reference-CNN = 0.15 were selected for subject 1, such that the Conv-CA can have a good performance on all the five tested time windows. And these hyperparameters were directly applied to other 34 subjects without further tuning. The hyperparameters were tuned in only one subject because that we hope the model is confident when applied on new subjects. Another reason is the huge computation cost while validating the entire dataset.

We tested other two hyperparameter sets on subject 1, 2, 3, 4 and 5 with $N_s = 50$ to demonstrate that the Conv-CA can be improved for a particular dataset by simply further tuning the hyperparameters. In Fig. 7, the hyperparameter set 1 reduced the dropout rate of signal-CNN from 0.75 to 0.65. The hyperparameter set 2 uses the short local time period = 12, number of filters in signal-CNN = 12, dropout rate of signal-CNN = 0.6, number of filters in reference-CNN = 40, and dropout rate of reference-CNN = 0.05. We chose a lower dropout rate because when $N_s = 50$, it has much less trainable parameters compare to $N_s = 250$. We increased the short local time period to enable the model to capture more time domain patterns. The original hyperparameters have 69.19% classification accuracy among these 5 subjects, while the hyperparameter set 1 reached 70.00% and the hyperparameter set 2 reached 70.64%. From Fig. 7, for subject 1, 2, 3, 4, and 5, the further tuned hyperparameters have better performances.

IV. Discussions

The TRCA-based methods learn weights of spatial filters from the training data to linear combine EEGs collected from multiple channels. While applying the ensemble TRCA with filter bank, the method has great performance on the benchmark dataset. However, for some subjects, for example the subject 11, 19, and 33, the classification accuracy is much lower than the cross-subject averaged accuracy. Specifically, the accuracies of the subject 11, 19, and 33 are 76.20%, 69.63%, and 57.78% compare to the averaged accuracy 92.99% on the 1.0 s time window. Our proposed Conv-CA model has significant better performance on these low-performed subjects.

The proposed Conv-CA model is no longer a linear model like spatial filters. It combines CNNs and correlation analysis to provide convolution operations for short periods of EEG signals across multiply channels. The Conv-CA suggests that by involving non-linear components, we can obtain a better performed model than spatial filters. It also implies that some subjects have worse SSVEP performances is probably because the model does not contain enough non-linear components. Furthermore, the CNN structures and parameters in the Conv-CA model has not been sufficiently explored. By constructing different CNNs, the Conv-CA can expand into different models. And the structure of the Conv-CA can also be further changed to the combinations of other non-linear structures and other signal processing tools.

The Conv-CA model is not a classic deep learning model which constructed by pure neural networks. It contains a correlation layer at the end of the CNNs. The correlation layer connects the signal-CNN and the reference-CNN with correlation coefficients. In other aspect, the CNNs are extensions of spatial filters, because the signals and the

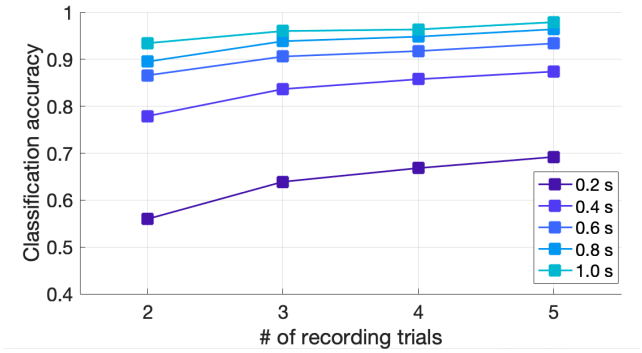

Fig. 6. The Conv-CA performance for subject 1, 2, 3, 4 and 5 under different number of training trials. When $N_s = 50$, there are 6,000 training trials in each recording trials. $N_s = 100, 150, 200$ and 250 are corresponding to 3040, 1880, 1400 and 1040 training trials in each recording trials.

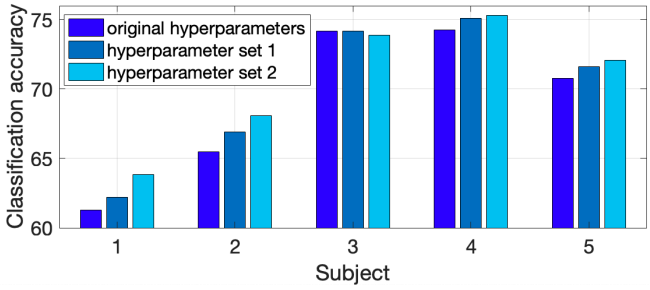

Fig. 7. The Conv-CA performance for subject 1, 2, 3, 4 and 5 and $N_s = 50$ on different hyperparameters. The hyperparameter set 1 has the short local time period = 9, number of filters in signal-CNN = 16, dropout rate of signal-CNN = 0.65, number of filters in reference-CNN = 40, and dropout rate of reference-CNN = 0.15. The hyperparameter set 2 uses the short local time period = 12, number of filters in signal-CNN = 12, dropout rate of signal-CNN = 0.6, number of filters in reference-CNN = 40, and dropout rate of reference-CNN = 0.05.

references are still connected with correlation analysis. Classic deep learning models learn all parameters from the dataset. Because EEG signals has low SNRs, it is easy to obtain a biased deep learning model, which means the model learns too much from the noise rather than focus on the signal. Even with huge amount of data, it will still be a challenge for deep learning models to locate the true signal. By involving a correlation layer, the CNNs are forced to maximize the signal-reference correlations. It can avoid bring too much redundant parameters and structures. The idea of combing traditional signal processing tools and deep learning structures can also be tested on other low SNR signals, such as P300 and seizure. Deep learning structures can bring sufficient non-linear operations to a model and the unsupervised features from traditional signal processing tools can limit the fitting space of the deep learning model.

Compare to ensemble TRCA, Conv-CA has much more computation cost on training. But in the testing process, the inference speed of Conv-CA is 0.6 ms per sample with a GPU, while the ensemble TRCA is 15 ms per sample. In structures, the first CNN layer of Conv-CA calculates frequency patterns in multiple filters. It is similar as the filter bank technique. The dense layer fuses the correlation coefficients in all frequencies together. It is similar as the ensemble technique. While in the TRCA implementation, it trained a set of weights for each EEG and reference pair. In other word, TRCA saved hundreds sets of weights such that each sampling time has a unique set of weights. But the Conv-CA only has one set of parameters for all the sampling time. So, a trained Conv-CA model stores much less parameter than TRCA.

For other works validated on dataset [26], CORCA [25] reported almost the same performance as ensemble TRCA, so we didn't perform extra study to compare with their work. PodNet [31] was validated on 1.5 s, 2 s, 5 s and 6 s time windows without applying sliding window. Although their implementation is different from TRCA and CORCA, its accuracy on 1.5 s time window is 75.64 %. The lowest accuracy among all the subjects is about 42%. The averaged accuracy of Conv-CA on 1.0 s time window is 93.88% with lowest accuracy 71.02%. Thus, the Conv-CA performs much better than the PodNet. The TSCORRCA model proposed by Zhang et al. [35] reached ITR 200 bit/min on 0.7 s time window. The best ITR of the ensemble TRCA model is 216 bit/min and the best ITR of the Conv-CA model is 224 bit/min. Wei et al. [36] proposed EN FBMSFA model has ITR 230 bit/min, which is higher than ensemble TRCA and Conv-CA. However, their result was validated on only one sample on each recording trial instead of using a sliding window to get multiple samples. Thus, it is not accurate for directly comparing their result with other studies.

V. Future Work

Our proposed Conv-CA has three potential improvement directions. First, because the correlation analysis eliminates the phase information, this model only uses the frequency information to classify SSVEPs. From this aspect, this model wastes some important information of the dataset. We believe the phase information should be extracted in a non-linear structure different from what we built to extract the frequency information. Adding phase information in the model has potential to further improve the current model. Second, the frequency patterns of the reference signals in the same class are not similar to each other all the time. This is because we pair each input EEG data sample with a reference signal sample. However, even though they have different phases, they should have a similar frequency patten. A more careful data preparation can also improve the model. Third, as the Fig. 6 demonstrated, the training data may not be sufficient for the model. Adjust the Conv-CA to make use of the data of other subjects should bring improvement to the Conv-CA. To validate the performance of our Conv-CA model, we are conducting on-line experiments in future work.

VI. Conclusion

In this study, we proposed a novel SSVEP classification model Conv-CA to classify multi-target SSVEPs using data from individual subjects. The Conv-CA model with a structure combining CNNs and correlation analysis significantly outperformed the state-of-the art method. Especially, the Conv-CA model improved the BCI performance on the subjects

whose SSVEP has lower performance. A benchmark dataset with thirty-five subjects and forty SSVEP frequencies were used to test the Conv-CA model. Experimental results indicated that the proposed model yields significantly higher classification accuracies and ITRs than the competing method. It also solves the problem that the current deep learning-based models are not completable with spatial filter methods.

VII. Acknowledgement

This work was supported by the National Science Foundation Award Number: 1464737. The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] Peng, Nengneng, et al. "Control of a nursing bed based on a hybrid brain-computer interface." *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016.
[2] Lo, Chi-Chun, Shang-Ho Tsai, and Bor-Shyh Lin. "Novel non-contact control system of electric bed for medical healthcare." *Medical & biological engineering & computing* 55.3 (2017): 517-526.
[3] Li, Yao, and Thenkurussi Kesavadas. "Brain computer interface robotic co-workers: Defective part picking system." *ASME 2018 13th International Manufacturing Science and Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2018.
[4] Li, Yao, and Thenkurussi Kesavadas. "Welding Robotic Co-Worker Using Brain Computer Interface." *ASME 2018 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers Digital Collection, 2018.
[5] Nakanishi, Masaki, et al. "Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis." *IEEE Transactions on Biomedical Engineering* 65.1 (2017): 104-112.
[6] Nguyen, Trung-Hau, Da-Lin Yang, and Wan-Young Chung. "A high-rate BCI speller based on eye-closed EEG signal." *IEEE Access* 6 (2018): 33995-34003.
[7] Chai, Xiaoke, et al. "A hybrid BCI-controlled smart home system combining SSVEP and EMG for individuals with paralysis." *Biomedical Signal Processing and Control* 56 (2020): 101687.
[8] Saboor, Abdul, et al. "SSVEP-based BCI in a smart home scenario." *International Work-Conference on Artificial Neural Networks*. Springer, Cham, 2017.
[9] Cruz, Inês, et al. "Kessel Run-A Cooperative Multiplayer SSVEP BCI Game." *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer, Cham, 2017.
[10] Chen, Shih-Chung, et al. "A Single-Channel SSVEP-Based BCI with a Fuzzy Feature Threshold Algorithm in a Maze Game." *International Journal of Fuzzy Systems* 19.2 (2017): 553-565.
[11] Koo, Bonkon, et al. "Immersive BCI with SSVEP in VR head-mounted display." *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2015.
[12] Chen, Xiaogang, et al. "Control of a 7-DOF robotic arm system with an SSVEP-based BCI." *International journal of neural systems* 28.08 (2018): 1850018.
[13] Chiuzbaian, Andrei, Jakob Jakobsen, and Sadasivan Puthusserypady. "Mind Controlled Drone: An Innovative Multiclass SSVEP based Brain Computer Interface." *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2019.
[14] Waytowich, Nicholas R., and Dean J. Krusienski. "Development of an extensible SSVEP-BCI software platform and application to wheelchair control." *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2017.
[15] Chen, Xiaogang, et al. "Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm." *Journal of neural engineering* (2018).
[16] Stawicki, Piotr, Felix Gembler, and Ivan Volosyak. "Driving a semiautonomous mobile robotic car controlled by an SSVEP-based BCI." *Computational intelligence and neuroscience*2016 (2016): 5.
[17] Regan, David. "Human brain electrophysiology." *Evoked potentials and evoked magnetic fields in science and medicine* (1989).
[18] Friman, Ola, Ivan Volosyak, and Axel Gräser. "Multiple channel detection of steady-state visual evoked potentials for brain-computer interfaces."*Biomedical Engineering, IEEE Transactions on* 54.4 (2007): 742-750.
[19] Y. Zhang et al., "Multiway canonical correlation analysis for frequency components recognition in SSVEP-based BCIs," in Proc. 18th Int. Conf. Neural Inform. Process., 2011, pp. 287–295.
[20] Y. Zhang et al., "L1-regularized multiway canonical correlation analysis for SSVEP-based BCI," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 21, no. 6, pp. 887–896, Nov. 2013.
[21] Y. Zhang et al., "Frequency recognition in SSVEP-based BCI using multi- set canonical correlation analysis," Int. J. Neural Syst., vol. 24, no. 4, 2014, Art. no. 1450013.
[22] Chen, Xiaogang, et al. "Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain–computer interface." Journal of neural engineering 12.4 (2015): 046008.
[23] Y. Wang et al., "Enhancing detection of steady-state visual evoked potentials using individual training data," in Proc. 36th Ann. Int. Conf. IEEE Eng. Med. Biol. Soc., 2014, pp. 3037–3040.
[24] Xu, Minpeng, Jin Han, Yijun Wang, Tzyy-Ping Jung, and Dong Ming. "Implementing over 100 command codes for a high-speed hybrid brain-computer interface using concurrent P300 and SSVEP features." *IEEE Transactions on Biomedical Engineering* (2020).
[25] Zhang, Yangsong, et al. "Correlated component analysis for enhancing the performance of SSVEP-based brain-computer interface." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26.5 (2018): 948-956.
[26] Wang, Yijun, et al. "A benchmark dataset for SSVEP-based brain–computer interfaces." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.10 (2016): 1746-1752.
[27] Gevins, Alan S., and N. H. Morgan. "Applications of neural-network (NN) signal processing in brain research." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 36, no. 7 (1988): 1152-1161.
[28] Liu, Qianqian, Yong Jiao, Yangyang Miao, Cili Zuo, Xingyu Wang, Andrzej Cichocki, and Jing Jin. "Efficient representations of EEG signals for SSVEP frequency recognition based on deep multiset CCA." *Neurocomputing*378 (2020): 36-44.
[29] Aznan, Nik Khadijah Nik, et al. "On the classification of SSVEP-based dry-EEG signals via convolutional neural networks." *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018.
[30] Kwak, No-Sang, Klaus-Robert Müller, and Seong-Whan Lee. "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment." *PloS one* 12.2 (2017): e0172578.
[31] Podmore, Joshua J., et al. "On the relative contribution of deep convolutional neural networks for SSVEP-based bio-signal decoding in BCI speller applications." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27.4 (2019): 611-618.
[32] Lawhern, Vernon J., Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces." *Journal of neural engineering* 15, no. 5 (2018): 056013.
[33] Wu, Hao, Fu Li, Yuchen Li, Boxun Fu, Guangming Shi, Minghao Dong, and Yi Niu. "A Parallel Multiscale Filter Bank Convolutional Neural Networks for Motor Imagery EEG Classification." *Frontiers in Neuroscience* 13 (2019): 1275.
[34] Wei, Chun-Shu, Toshiaki Koike-Akino, and Ye Wang. "Spatial component-wise convolutional network (SCCNet) for motor-imagery EEG classification." In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 328-331. IEEE, 2019.
[35] Zhang, Yangsong, Erwei Yin, Fali Li, Yu Zhang, Toshihisa Tanaka, Qibin Zhao, Yan Cui, Peng Xu, Dezhong Yao, and Daqing Guo. "Two-stage frequency recognition method based on correlated component analysis for SSVEP-based BCI." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 26, no. 7 (2018): 1314-1323.
[36] Wei, Qingguo, Shan Zhu, Yijun Wang, Xiaorong Gao, Hai Guo, and Xuan Wu. "Maximum signal fraction analysis for enhancing signal-to-noise ratio of EEG signals in SSVEP-based BCIs." *IEEE Access* 7 (2019): 85452-85461.