Ensemble of Convolutional Neural Networks for P300 Speller in Brain Computer Interface

Hongchang Shan[1(✉)], Yu Liu[2], and Todor Stefanov[1]

[1] Leiden University, Leiden, The Netherlands
{h.shan,t.p.stefanov}@liacs.leidenuniv.nl
[2] KU Leuven, Leuven, Belgium
yu.liu@esat.kuleuven.be

Abstract. A Brain Computer Interface (BCI) speller allows human-beings to directly spell characters using eye-gazes, thereby building communication between the human brain and a computer. Convolutional Neural Networks (CNNs) have shown better ability than traditional machine learning methods to increase the character spelling accuracy for the BCI speller. Unfortunately, current CNNs can not learn well the features related to the target signal of the BCI speller. This issue limits these CNNs from further character spelling accuracy improvements. To address this issue, we propose a network, which combines our proposed two CNNs, with an existing CNN. These three CNNs of our network extract different features related to the target BCI signal. Our network uses the ensemble of the features extracted by these CNNs for BCI character spelling. Experimental results on three benchmark datasets show that our network outperforms other methods in most cases, with a significant spelling accuracy improvement up to 38.72%. In addition, the communication speed of the P300 speller based on our network is up to 2.56 times faster than the communication speed of the P300 speller based on other methods.

1 Introduction

A Brain Computer Interface (BCI) enables direct communication between the human brain and a computer by analyzing the human's neural activities. In this way, human-beings can use only the brain to express their thoughts without any real movement. Traditionally, BCIs are conceived as a pathway for people suffering from motor disabilities [10]. With the rapid development of BCIs, recent research is also focused on developing BCIs for healthy users to allow users' hands-free interaction with applications such as games [3], mental state monitoring [14], and IoT services [13]. Due to their non-invasiveness, easiness and safety, Electroencephalogram (EEG)-based BCIs attract most of the research. Among all kinds of EEG-based BCIs, the P300 speller is one of the most-commonly investigated applications because the P300 speller has a good performance on

I. V. Tetko et al. (Eds.): ICANN 2019, LNCS 11730, pp. 376–394, 2019.
https://doi.org/10.1007/978-3-030-30490-4_31

character spelling [10]. Therefore, this paper considers the P300 speller as our target BCI application.

Previously, traditional machine learning methods were used for character spelling in the P300 speller. These methods employ signal processing techniques for feature extraction and use classifiers such as Support Vector Machine (SVM) or Linear Discriminant Analysis (LDA) for the detection of P300 signals and the inference of characters. For example, Rivet [22] enhances the P300 potentials. Mennes [17] removes artifacts in the EEG recordings containing P300 signals. Bostanov [4] extracts useful features related to P300 signals. However, there are some problems with traditional machine learning methods. (1) they can only learn the features that researchers are focusing on but lose or remove other underlying features [23]; (2) brain signals have subject-to-subject variability, which makes it possible that methods performing well on certain subjects (with similar age or occupation) may not give a satisfactory performance on others. These problems prevent traditional machine learning methods from further increasing the character spelling accuracy for the P300 speller.

In recent years, deep learning, especially deep Convolutional Neural Networks (CNNs), has achieved significant success in the computer vision field. CNNs have the advantage of automatically learning features from raw data[1]. They can learn not only something we know but also something important and unknown to us [6,23]. Automatically learning from raw data has better ability to achieve good results which are invariant to different subjects. Thus, CNNs are able to boost the full potential of detecting BCI signals, overcoming the aforementioned shortcomings of traditional machine learning methods.

Therefore, in recent years, researchers have started to design (deep) CNNs for P300-based BCIs [6,15,16,23]. However, these CNNs have some limitations in increasing the P300 spelling accuracy. CNNs in [6,15,16] first use a spatial convolution layer to learn P300-related spatial features from raw signals. Then, they use several temporal convolution layers to learn P300-related temporal features from the abstract signals generated by the spatial convolution layer (the first layer). The abstraction of raw signals loses raw temporal information, which makes these CNNs not able to learn P300-related temporal features well. To solve the problem of [6,15,16], the CNN in [23] performs the spatial convolution and the temporal convolution at the same time (thereby performing the spatial-temporal convolution) in the first layer. The input to the first layer is raw signals. Thus, the CNN in [23] is able to learn temporal features from raw signals instead of abstract signals as in [6,15,16]. In this way, [23] learns better P300-related temporal features than [6,15,16]. Unfortunately, [23] extracts only P300-related joint spatial-temporal features through the spatial-temporal convolution. It does not extract P300-related separate temporal features and separate spatial features. These separate temporal features and separate spatial features have proven to be very important for the P300 speller [9,11,19,20]. Adding

[1] In this paper, we use "raw data, information, or signals" to denote the data which are only preprocessed (e.g., bandpass filtering and normalization) but not abstracted by a feature extraction method (e.g., a CNN).

several temporal or spatial convolution layers after the first spatial-temporal convolution layer enables [23] to learn P300-related separate spatial or temporal features. Nevertheless, this cannot make [23] learn these features well because the input to these added temporal or spatial convolution layers is the abstract signals generated by the first spatial-temporal convolution layer instead of raw signals. This leads to the loss of raw information related to the P300 signal. In order to solve this issue in [23], we propose a network which combines our proposed two CNNs, with the CNN in [23] for character spelling in the P300 speller. The novel contributions of this paper are the following:

- Each of our proposed two CNNs has only one convolution layer. One of the CNNs performs the temporal convolution in the convolution layer (the first layer) to extract P300-related separate temporal features. The other CNN performs the spatial convolution in the convolution layer (the first layer) to extract P300-related separate spatial features. These two CNNs are able to learn well P300-related separate temporal features and separate spatial features, respectively.
- Experimental results on three benchmark datasets show that our network, which is the ensemble of our two CNNs and OCLNN [23], outperforms other methods in most cases, with a significant spelling accuracy improvement up to 38.72%. In addition, the communication speed of the P300 speller based on our network is up to 2.56 times faster than the communication speed of the P300 speller based on other methods.

The rest of the paper is organized as follows: Sect. 2 describes the related work on P300 spelling. Section 3 introduces some background information about the P300 speller, and the datasets used in this paper. Section 4 presents our proposed network for P300 spelling. Section 5 compares the character spelling accuracy and the communication speed achieved by our network and other methods for the P300 speller. Section 6 analyses our proposed two CNNs on extracting P300-related features, performs an ablation study on our proposed network and discusses the importance of extracting P300-related features from raw signals. Section 7 ends the paper with conclusions.

2 Related Work

In [6,16], and [15], the authors propose CNNs for character spelling in the P300 speller. The CNN in [6,16], and [15] is called CCNN [6], CNN-R [16], and BN3 [15], respectively. CCNN, CNN-R, and BN3 first use a spatial convolution layer to learn P300-related spatial features. After this spatial convolution layer, they use several temporal convolution layers to learn P300-related temporal features. However, the problem of these CNNs is that they learn P300-related temporal features from abstract signals instead of raw signals, which makes these CNNs not able to learn P300-related temporal features well. P300-related temporal features are learned by the temporal convolution layers of these CNNs. The input to these temporal convolution layers is the feature maps generated by

the spatial convolution layer (the first layer). These feature maps are abstract temporal signals instead of raw signals because this spatial convolution layer converts each receptive field of raw signals into an abstract datum in a feature map. These abstract temporal signals in the feature maps lose raw temporal information. Losing raw temporal information means losing important temporal features because the nature of P300 signals is the positive voltage potential in raw temporal information, see Fig. 1 explained in Sect. 3.1, as well as many important P300-related features are also embodied in raw information [20,23]. As a result, these CNNs can not learn temporal features well and can not further increase the spelling accuracy of the P300 speller.

In order to solve the problem of [6,16], and [15,23] proposes a CNN with one convolution layer, called OCLNN, for character spelling in the P300 speller. In contrast to CCNN [6], CNN-R [16], and BN3 [15], the network OCLNN [23] performs the spatial convolution and the temporal convolution at the same time, thereby performing the spatial-temporal convolution in the first layer instead of performing only the spatial convolution as in CCNN, CNN-R, and BN3. The input to this spatial-temporal convolution layer (the first layer) is raw signals. In this way, the data used to learn P300-related temporal features is raw signals instead of the abstract signals in CCNN, CNN-R, and BN3. Therefore, OCLNN is able to learn P300-related temporal features better than CCNN, CNN-R, and BN3. In addition, OCLNN can learn spatial features. As a result, OCLNN achieves higher spelling accuracy than CCNN, CNN-R, and BN3. Unfortunately, OCLNN loses other important P300-related features. OCLNN extracts P300-related spatial and temporal features at the same time in its single convolution layer, thereby extracting only P300-related joint spatial-temporal features through the spatial-temporal convolution. OCLNN does not extract P300-related separate temporal features and separate spatial features. These separate temporal features and separate spatial features have proven to be very important for the P300 speller [9,11,19,20]. Adding several temporal or spatial convolution layers after the first spatial-temporal convolution layer is a potential method to enable OCLNN to learn P300-related separate spatial or temporal features. Nevertheless, this method can not learn P300-related separate temporal or spatial features well due to the loss of raw information. The raw information loss happens because the input to these added temporal or spatial convolution layers for OCLNN is the abstract signals (generated by the first spatial-temporal convolution layer in OCLNN) instead of raw signals.

To address this issue of [23], we proposes a network which combines our proposed two CNNs with OCLNN in order to learn well the aforementioned P300-related separate spatial and temporal features, which are not extracted by OCLNN, as well as the spatial-temporal features. Each of these two CNNs has only one convolution layer. One of the CNNs performs the temporal convolution in the first layer to learn P300-related separate temporal features. The other CNN performs the spatial convolution in the first layer to learn P300-related separate spatial features. In this way, the input to each of the two CNNs is raw signals, thus these two CNNs are able to learn features from raw

signals instead of the abstract signals in the aforementioned potential method for enabling OCLNN to learn more features. As a consequence, these two CNNs can learn well P300-related separate temporal features and separate spatial features, respectively. Our network uses the ensemble of these two CNNs and OCLNN, thereby extracting more useful P300-related features than OCLNN. As a result, our proposed network can achieve higher spelling accuracy than OCLNN.

3 Background

In this section, we provide some background information for the P300 speller and the benchmark datasets used in this paper.

3.1 P300 Speller

The P300 speller is one of the most investigated applications in BCI [10]. A target character is spelled using the property of the P300 signal. As shown in Fig. 1, a P300 signal, recorded in EEG, occurs as a positive deflection in voltage with a latency of about 300 ms after a rare stimulus is presented to a subject (person). The following experiment is used to evoke a P300 signal in a subject's brain and then the evoked P300 signal is used to spell characters. In this experiment, the subject is presented with a 6 by 6 character matrix (see Fig. 2) and he focuses his attention on a target character he wants to spell. The matrix performs random, separate, and successive row or column intensification. When the target row or column is intensified, it is a rare stimulus to the subject because there are only two target intensifications out of 12 intensifications. This rare stimulus evokes the subject's brain to generate a P300 signal. Then, with the detection of a P300 signal, the target row or column is inferred. By combining the target row position and the target column position, the target character position is inferred. Assume that one epoch includes 12 intensifications, in which there exist one target row intensification and one target column intensification. In practice, people use several consecutive epochs for the P300 speller to infer

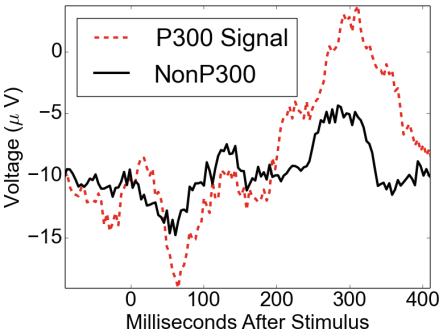

Fig. 1. P300 signal.

Fig. 2. P300 speller character matrix.

one target character, because it is hard to use only one epoch to correctly spell one target character [21,23].

3.2 Datasets

We perform experiments on three benchmark datasets, i.e., BCI Competition II - Data set IIb [1] as well as BCI Competition III - Data set II Subject A and Subject B [2]. In this paper, we use II to represent BCI Competition II - Data set IIb, III-A to represent BCI Competition III - Data set II Subject A, and III-B to represent BCI Competition III - Data set II Subject B. These three benchmark datasets are commonly used to evaluate many methods for the P300 speller [4,6,15,16,21,23]. Therefore, we are able to fairly compare the spelling accuracy achieved by our proposed network and other state-of-the-art methods for the P300 speller.

In Dataset II, III-A and III-B, the EEG signals are recorded from 64 sensors at a sampling frequency of 240 Hz when performing the P300 speller experiment described in Sect. 3.1. In this P300 speller experiment, one row or column is intensified for 100 ms. After each row/column intensification, the matrix is blank for 75 ms. In this experiment, 15 consecutive epochs are used for the spelling of one character. After every group of 15 epochs, the matrix is blank for 2.5 s to inform the subject to focus on the next character to spell.

In Dataset II, III-A and III-B, there are separate training and test datasets. In Dataset II, the training dataset has 42 characters and the test dataset has 31 characters. Since 15 epochs are used for the spelling of one character, the total number of epochs is 630 epochs and 465 epochs in the training dataset and test dataset, respectively. The training dataset in Dataset III-A and the training dataset in Dataset III-B have the same number of characters, i.e., 85 characters. The test dataset in Dataset III-A and the test dataset in Dataset III-B also have the same number of characters, i.e., 100 characters. Therefore, in Dataset III-A and Dataset III-B, the total number of epochs is 1275 epochs and 1500 epochs in each training dataset and each test dataset, respectively.

4 Proposed Network

This section introduces our proposed network for character spelling in the P300 speller. We call our network Ensemble of Convolutional Neural Networks (EoCNN). EoCNN uses our proposed two CNNs. We call these two CNNs One Spatial Layer Network (OSLN) and One Temporal Layer Network (OTLN).

4.1 Ensemble of Convolutional Neural Networks

The workflow of our EoCNN is shown in Fig. 3. First, the EEG signals are preprocessed to construct the input tensor. The construction of the input tensor is described in Sect. 4.2. Then, the input tensor is sent to three different CNNs, i.e., OSLN, OTLN, and OCLNN. OSLN and OTLN are described in

Sect. 4.3. OCLNN is the CNN proposed in [23]. OSLN extracts P300-related separate spatial features. OTLN extracts P300-related separate temporal features. OCLNN extracts P300-related joint spatial-temporal features. Our EoCNN uses the ensemble of the outputs from OSLN, OTLN, and OCLNN for character spelling in the P300 speller.

4.2 Input Tensor

The EEG signals are preprocessed to construct the input tensor ($Tem \times C$), where C is the number of sensors used to acquire EEG signals. Tem is the number of signal samples in the time domain. In this tensor, in order to remove the high frequency noise, the temporal signal samples are bandpass filtered between 0.1 Hz and 20 Hz. Then, we normalize the temporal signal samples to make the signal samples to have zero mean and unit variance based on each individual pattern and for each sensor. Here an individual pattern denotes the Tem signal samples. The normalization is a common practice for preprocessing input data to CNNs. The normalization helps the CNN to perform well for the P300 spelling [6].

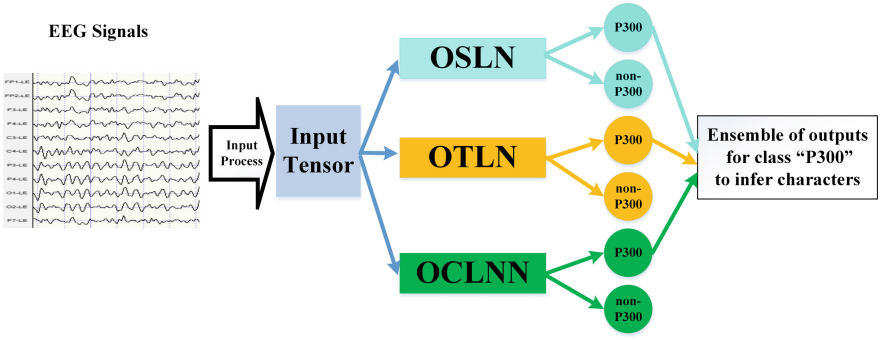

Fig. 3. Workflow of our EoCNN

4.3 Proposed OSLN and OTLN

The architectures of our proposed OSLN and OTLN are described in Tables 1 and 2, respectively. OSLN and OTLN are used in EoCNN (see Sect. 4.1), where OSLN is designed to learn P300-related separate spatial features and OTLN is designed to learn P300-related separate temporal features. Since only the convolution layer is different between OSLN and OTLN, below we describe the architectures of OSLN and OTLN together.

Layer 1 of OSLN (see Table 1) performs the spatial convolution operation with the kernel size (1, C). This convolution operation converts each receptive field of the signal samples into an abstract datum in a feature map. The signal samples in each receptive field are from all C sensors in the space domain and

Table 1. OSLN architecture.

Layer	Operation	Kernel	Feature maps or neurons
1	Convolution	$(1, C)$	16
	Dropout	—	—
2	Fully-Connected	—	2

Table 2. OTLN architecture.

Layer	Operation	Kernel	Feature maps or neurons
1	Convolution	$(Tem/15, 1)$	16
	Dropout	—	—
2	Fully-Connected	—	2

sampled at only one time point in the time domain. Therefore, this convolution operation extracts P300-related separate spatial features. We use the kernel size $(1, C)$ in order to make this layer to learn the spatial features from EEG signals acquired using all sensors. The reason for using all sensors is that it is more helpful to increase the spelling accuracy than using part of all sensors [6,15,16,23]. The input to this layer is the raw signals, so this layer learns P300-related separate spatial features from raw signals. This layer generates 16 feature maps, which are the input to Layer 2 of OSLN. The choice of 16 feature maps follows the suggestion in [23].

Layer 1 of OTLN (see Table 2) performs the temporal convolution operation with the kernel size $(Tem/15, 1)$. The temporal convolution operation converts each receptive field of the signal samples into an abstract datum in a feature map. The signal samples in each receptive field are sampled within a certain time period and are acquired from only one sensor. Therefore, this convolution operation extracts P300-related separate temporal features. We use the kernel size $(Tem/15, 1)$ because $1/15$ of temporal signal samples is a proper receptive field for a CNN to learn P300-related temporal features [23]. The input to this layer is the raw signals, so this layer learns P300-related separate temporal features from raw signals. This layer generates also 16 feature maps, which are the input to Layer 2 of OTLN.

In both Layer 1 of OSLN and Layer 1 of OTLN, the activation function is the Rectified Linear Unit (ReLU) [18] function. We employ dropout [25], with a rate of 0.4, to prevent OSLN and OTLN from overfitting.

Layer 2 of OSLN (see Table 1) and Layer 2 of OTLN (see Table 2) are the same. This layer is a fully-connected layer with two neurons. These two neurons represent the class "P300" (the presence of a P300 signal) and the class "non-P300" (the absence of a P300 signal), respectively. The activation function used in this layer is the Softmax [12] function which outputs the predicted probability for the "P300" class and the "non-P300" class.

OSLN and OTLN each uses only one convolution layer. OSLN uses only one convolution layer because it does not make sense to add more spatial convolution layers for OSLN. This CNN is designed to learn P300-related spatial features from the EEG signals recorded with all C sensors in the first layer. If we add more spatial convolution layers after its first spatial convolution layer to learn P300-related spatial features, these added layers should learn spatial features

from the abstract signals generated by the first spatial convolution layer. These abstract signals include only the time domain and do not have the space domain because the first convolution layer uses the receptive field including all C sensors. Thus, these abstract signals can not be used to extract spatial features. OTLN also uses only one convolution layer because one convolution layer is enough to extract useful P300-related separate temporal features (see Sect. 6.1).

4.4 Training

The training is carried out by minimizing the binary cross-entropy loss function [8]. It uses a Stochastic Gradient Descent [5] optimizer with momentum and weight decay. The momentum is 0.9 and the weight decay is 0.0005. The learning rate is fixed to 0.01. The batch size is 128. This setup of the training parameters follows the suggestion in [24].

4.5 Character Spelling Using EoCNN

The character spelling approach using EoCNN is performed by Eqs. (1), (2), (3), and (4).

$$P_{EoC}(i,j) = \frac{1}{3} \times (P_{OS}(i,j) + P_{OT}(i,j) + P_{OCL}(i,j)) \tag{1}$$

$$Sum_{(j)} = \sum_{i=1}^{k} P_{EoC}(i,j) \tag{2}$$

$$index_{col} = \underset{1 \le j \le 6}{argmax}\ Sum_{(j)} \tag{3}$$

$$index_{row} = \underset{7 \le j \le 12}{argmax}\ Sum_{(j)} \tag{4}$$

Equation (1) shows the ensemble processing of the outputs from OSLN, OTLN, and OCLNN. The output from a CNN used for character spelling is the predicted probability by this CNN for class "P300". In this equation, for epoch i and for intensification j, $P_{OS}(i,j)$ denotes the predicted probability by OSLN for class "P300", $P_{OT}(i,j)$ denotes the predicted probability by OTLN for class "P300", and $P_{OCL}(i,j)$ denotes the predicted probability by OCLNN for class "P300".

The calculation for the position of the target character when using the first k epochs is defined by Eqs. (2), (3) and (4), where $Sum_{(j)}$ denotes the sum of the predicted probabilities by EoCNN, $index_{col}$ denotes the index of the column position of the target character, and $index_{row}$ denotes the index of the row position of the target character. j denotes a column intensification when $j \in [1,6]$ and j denotes a row intensification when $j \in [7,12]$.

5 Experimental Evaluation

First, we introduce our experimental setup in Sect. 5.1. Then, we compare the spelling accuracy achieved by our EoCNN and other methods in Sect. 5.2. Finally, we compare the communication speed of the P300 speller based on our EoCNN and other methods in Sect. 5.3.

5.1 Experimental Setup

We use Keras with the Tensorflow backend [7] to implement our EoCNN.

We use every training dataset of Dataset II, III-A and III-B to train our EoCNN, separately. Therefore, for the input tensor to EoCNN (see Sect. 4.2), we have $C = 64$ because the number of sensors used to acquire EEG signals is 64. We have $Tem = 240$ because we take each individual pattern as the signals from the time period between 0 and 1000 ms posterior to the beginning of each intensification, and the signal sampling frequency is 240 Hz.

We evaluate every trained EoCNN on the corresponding test dataset of Dataset II, III-A and III-B, and calculate the spelling accuracy for every test dataset. The spelling accuracy is calculated using Eq. (5), where acc_k denotes the spelling accuracy when using the first k epochs for every character, R_k denotes the number of correctly inferred characters when using the first k epochs for every character, and A denotes the number of all characters.

$$acc_k = \frac{R_k}{A} \tag{5}$$

We compare our EoCNN with CCNN [6], BN3 [15], CNN-R [16], OCLNN [23], and Bostanov [4] on Dataset II. CCNN, BN3, CNN-R, and OCLNN are different CNNs used for the character spelling in the P300 speller. Bostanov is the method which won the championship on Dataset II in the BCI Competition II. We compare our EoCNN with CCNN, BN3, CNN-R, OCLNN, and ESVM [21] on Dataset III-A and Dataset III-B. ESVM is the method which won the championship on Dataset III-A and Dataset III-B in the BCI Competition III.

5.2 Character Spelling Accuracy

The spelling accuracy achieved by our EoCNN and other methods on Dataset II, Dataset III-A, and Dataset III-B is shown in Tables 3, 4, and 5, respectively. In these tables, the different methods, we compare, are shown in the first column. The spelling accuracy for different epoch numbers $k \in [1, 15]$ is shown in each row of the table. A number in bold indicates that the accuracy achieved by the corresponding method is the highest among all methods. "–" denotes that the corresponding paper, describing the method, does not provide this accuracy number. The accuracy in this table is shown in %. Overall, the spelling accuracy achieved by our EoCNN is higher than the spelling accuracy achieved by other methods in most cases. Our EoCNN increases the spelling accuracy achieved by other methods with up to 38.72%.

Table 3 shows that for Dataset II, for every epoch number $k \in [1, 15]$, the spelling accuracy achieved by our EoCNN is higher than the spelling accuracy achieved by all other methods. Our EoCNN can increase the spelling accuracy achieved by CCNN, CNN-R, BN3, OCLNN, and Bostanov with up to 38.72%, 12.90%, 19.36%, 6.45%, and 19.35%, respectively.

Table 4 shows that for Dataset III-A, in 14 out of 15 cases (epoch number $k \in [1, 8] \cup [10, 15]$), the spelling accuracy achieved by our EoCNN is higher than the spelling accuracy achieved by all other methods. Our EoCNN can increase

Table 3. Spelling accuracy achieved by different methods on Dataset II.

Method	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN	**83.87**	**93.55**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
CCNN	58.06	54.83	77.41	93.54	93.54	93.54	93.54	96.77	96.77	**100**	**100**	**100**	**100**	**100**	**100**
CNN-R	70.97	83.87	93.55	96.77	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
BN3	77.42	74.19	80.65	83.87	93.55	96.77	96.77	96.77	**100**	**100**	**100**	**100**	**100**	**100**	**100**
OCLNN	77.42	90.32	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
Bostanov	64.52	83.87	93.55	96.77	96.77	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

Table 4. Spelling accuracy achieved by different methods on Dataset III-A.

Method	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN	**23**	**39**	**61**	**68**	**76**	**81**	**84**	**86**	88	**93**	**95**	**98**	**97**	**99**	**99**
CCNN	16	33	47	52	61	65	77	78	85	86	90	91	91	93	97
CNN-R	14	28	38	53	57	62	71	75	77	82	89	87	87	92	95
BN3	22	**39**	58	67	73	75	79	81	82	86	89	92	94	96	98
OCLNN	**23**	**39**	56	63	73	79	82	85	**90**	91	94	95	95	96	**99**
ESVM	16	32	52	60	72	–	–	–	–	83	–	–	94	–	97

Table 5. Spelling accuracy achieved by different methods on Dataset III-B.

Method	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN	**51**	**66**	**74**	**81**	**84**	**90**	**91**	92	**95**	**97**	**98**	**98**	**98**	**98**	**99**
CCNN	35	52	59	68	79	81	82	89	92	91	91	90	91	92	92
CNN-R	36	46	66	70	77	80	86	86	88	91	94	95	95	96	96
BN3	47	59	70	73	76	82	84	91	94	95	95	95	94	94	95
OCLNN	46	62	72	79	**84**	87	89	**93**	94	96	97	97	97	**98**	98
ESVM	35	53	62	68	75	–	–	–	–	91	–	–	96	–	96

the spelling accuracy achieved by CCNN, CNN-R, BN3, OCLNN, and ESVM with up to 16%, 23%, 7%, 5%, and 10%, respectively.

Table 5 shows that for Dataset III-B, in 14 out of 15 cases (epoch number $k \in [1, 7] \cup [9, 15]$), the spelling accuracy achieved by our EoCNN is higher than the spelling accuracy achieved by all other methods. Our EoCNN can increase the accuracy achieved by CCNN, CNN-R, BN3, OCLNN, and ESVM with up to 16%, 20%, 8%, 5%, and 16%, respectively.

Moreover, our method is robust across different subjects. Tables 3, 4, and 5 show that for all three subjects, our EoCNN achieves the highest spelling accuracy among all other methods in 43 out of 45 cases.

These experimental results also give some insights on how many epochs we should use for the spelling of one character in the P300 speller. The first insight is from the fact that, in Table 3, the spelling accuracy achieved by CCNN and BN3 on epoch number $k = 2$ is lower than the spelling accuracy achieved by CCNN and BN3 on epoch number $k = 1$. This shows that adding more epochs does not necessarily improve spelling accuracy for the P300 speller. This is also discussed in more details in [6]. The other insight is from the fact that in Dataset II, we need only 2 epochs to achieve the spelling accuracy which is higher than 90% while in Dataset III-A and Dataset III-B, in order to achieve the spelling accuracy higher than 90%, we need at least 10 epochs and 6 epochs, respectively. This indicates us that we can use different number of epochs for different subjects to spell characters using the P300 speller. In this way, we can use a small number of epochs for a subject when using the P300 speller such that we can significantly decrease the time needed for a subject to spell a character while keeping an acceptable spelling accuracy.

5.3 Information Transfer Rate

This section compares the Information Transfer Rate (ITR) of the P300 speller based on our EoCNN and other methods. ITR has been the most commonly applied metric to assess the communication speed of BCIs [26], combining the accuracy and the time needed for recognition. It is calculated by Eqs. (6) and (7) [27], where P denotes the probability to correctly spell a character, N denotes the number of classes, and T denotes the time needed to spell a character when using k epochs. For more detailed explanation about the ITR please refer to [27].

$$ITR = \frac{60(P\log_2(P) + (1-P)\log_2\frac{1-P}{N-1} + \log_2(N))}{T} \tag{6}$$

$$T = 2.5 + 2.1k \quad 1 \le k \le 15 \tag{7}$$

The ITR of the P300 speller based on our EoCNN and other methods for Dataset II, Dataset III-A and Dataset III-B is shown in Tables 6, 7, and 8, respectively. In these tables, the different methods, we compare, are shown in the first column. The ITR for different epoch numbers $k \in [1, 15]$ is shown in each row of the table. A number in bold denotes that the corresponding method achieves the highest ITR for the P300 speller, compared with all other

methods. "–" in a table denotes that the ITR cannot be calculated because the corresponding method does not provide the spelling accuracy. The ITR is shown in bits/minute. Overall, in 43 out of 45 cases, the ITR of the P300 speller based on our EoCNN is higher than the ITR of the P300 spellers based on all other methods. The communication speed (i.e., ITR) of the P300 speller based on our EoCNN is up to 2.56 times faster than the communication speed of the P300 speller based on other methods.

Table 6. The ITR of the P300 speller based on different methods on Dataset II.

Method	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN	**88.92**	**58.62**	**46.28**	**35.24**	**28.45**	**23.85**	**20.53**	**18.03**	**16.07**	**14.49**	**13.19**	**12.11**	**11.19**	**10.41**	**9.72**
CCNN	48.9	24.26	29.02	30.63	24.73	20.74	17.85	16.74	14.92	**14.49**	**13.19**	**12.11**	**11.19**	**10.41**	**9.72**
CNN-R	67.48	48.33	40.25	32.72	**28.45**	**23.85**	**20.53**	**18.03**	**16.07**	**14.49**	**13.19**	**12.11**	**11.19**	**10.41**	**9.72**
BN3	77.79	39.42	31.06	25.26	24.74	22.15	19.07	16.74	**16.07**	**14.49**	**13.19**	**12.11**	**11.19**	**10.41**	**9.72**
OCLNN	77.79	54.97	**46.28**	**35.24**	**28.45**	**23.85**	**20.53**	**18.03**	**16.07**	**14.49**	**13.19**	**12.11**	**11.19**	**10.41**	**9.72**
Bostanov	57.88	48.33	40.25	32.72	26.41	**23.85**	**20.53**	**18.03**	**16.07**	**14.49**	**13.19**	**12.11**	**11.19**	**10.41**	**9.72**

Table 7. The ITR of the P300 speller based on different methods on Dataset III-A.

Method	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN	**10.62**	**14.04**	**19.74**	**17.89**	**17.31**	**16.13**	**14.76**	**13.49**	12.51	**12.46**	**11.81**	**11.55**	**10.44**	**10.14**	**9.48**
CCNN	5.45	10.67	13.02	11.65	12.14	11.26	12.76	11.45	11.78	10.84	10.69	10.01	9.25	8.95	9.07
CNN-R	4.19	8.11	9.24	12.01	10.89	10.44	11.18	10.73	9.99	10	10.48	9.25	8.55	8.77	8.7
BN3	9.81	**14.04**	18.22	17.47	16.2	14.2	13.32	12.19	11.09	10.84	10.48	10.21	9.82	9.51	9.27
OCLNN	**10.62**	**14.04**	17.22	15.83	16.2	15.47	14.17	13.22	**13.02**	11.98	11.58	10.84	10.02	9.51	**9.48**
ESVM	5.45	10.14	15.3	14.64	15.84	–	–	–	–	10.21	–	–	9.82	–	9.07

Table 8. The ITR of the P300 speller based on different methods on Dataset III-B.

Method	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN	**39.76**	**32.62**	**26.95**	**23.82**	**20.45**	**19.33**	**16.97**	15.2	**14.38**	**13.52**	**12.58**	**11.55**	**10.67**	**9.92**	**9.48**
CCNN	21.64	22.29	18.72	17.89	18.45	16.13	14.17	14.32	13.55	11.98	10.91	9.82	9.25	8.77	8.2
CNN-R	22.67	18.32	22.4	18.75	17.68	15.79	15.37	13.49	12.51	11.98	11.58	10.84	10.02	9.51	8.88
BN3	34.9	27.27	24.63	20.07	17.31	16.46	14.76	14.9	14.1	12.97	11.81	10.84	9.82	9.13	8.7
OCLNN	33.71	29.51	25.78	22.85	**20.45**	18.21	16.31	**15.51**	14.1	13.24	12.31	11.3	10.44	**9.92**	9.27
ESVM	21.64	22.98	20.26	17.89	16.93	–	–	–	–	11.98	–	–	10.23	–	8.88

Table 6 shows that the communication speed of the P300 speller based on our EoCNN is up to 2.42 times faster than the communication speed of the P300 speller based on other methods. The maximum increase of the communication speed occurs when comparing the ITR of the P300 speller based on our EoCNN with the ITR of the P300 speller based on CCNN for epoch number $k = 2$.

Table 7 shows that the communication speed of the P300 speller based on our EoCNN is up to 2.56 times faster than the communication speed of the P300 speller based on other methods. The maximum increase of the communication speed occurs when comparing the ITR of the P300 speller based on our EoCNN with the ITR of the P300 speller based on CNN-R for epoch number $k = 1$.

Table 8 shows that the communication speed of the P300 speller based on our EoCNN is up to 1.84 times faster than the communication speed of the P300 speller based on other methods. The maximum increase of the communication speed occurs when comparing the ITR of the P300 speller based on our EoCNN with the ITR of the P300 speller based on CCNN and ESVM for epoch number $k = 1$.

These experimental results show that by using our EoCNN, the communication speed of the P300 speller can be significantly increased for low epoch numbers.

6 Discussions

In this section, first, we analyse our proposed OTLN and OSLN in terms of spelling accuracy, and discuss the influence of the number of convolution layers on extracting useful P300-related separate temporal features. Then, we perform an ablation study on EoCNN. Finally, we explore the importance of extracting P300-related temporal features from raw signals.

In this section, all the experiments are performed by using the experimental setup described in Sect. 5.1. We have done experiments using all three datasets, which show the similar conclusions. Thus, we only present the experimental results on Dataset III-A.

6.1 Analysis on Our Proposed OTLN and OSLN

First, we perform experiments to show the spelling accuracy achieved by OTLN and OSLN, respectively. The experimental results are shown in Table 9. In this table, different CNNs, we compare, are shown in the first column. The spelling accuracy for different epoch numbers $k \in [1, 15]$ is shown in each row of the table. A number in bold indicates that the corresponding CNN achieves the highest accuracy compared to all other CNNs. The accuracy in this table is shown in %. Table 9 shows that OTLN and OSLN both have good ability to achieve high spelling accuracy when OTLN and OSLN are used independently for P300 spelling. Thus, OTLN and OSLN are able to extract very useful P300-related separate temporal features and P300-related separate spatial features, respectively.

Then, we analyse whether OTLN needs more convolution layers to extract P300-related separate temporal features. In order to analyse the influence of the number of convolution layers on OTLN, we perform experiments to compare the spelling accuracy achieved by OTLN and other two CNNs called OTLN-3l and OTLN-6l. OTLN-3l and OTLN-6l use 3 and 6 convolution layers, respectively.

Table 9. Spelling accuracy achieved by OTLN, OSLN and EoCNN on Dataset III-A.

Network	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OTLN	21	34	51	65	69	73	76	81	85	85	88	92	92	93	95
OSLN	**24**	35	55	63	69	75	78	79	80	82	89	92	94	95	96
EoCNN	23	**39**	**61**	**68**	**76**	**81**	**84**	**86**	**88**	**93**	**95**	**98**	**97**	**99**	**99**

These convolution layers use the same kernel size and generate the same number of feature maps as the convolution layer used in OTLN. The spelling accuracy achieved by OTLN, OTLN-3l and OTLN-6l is plotted in Fig. 4. This figure shows that the spelling accuracy achieved by OTLN-3l and OTLN is almost the same. The spelling accuracy achieved by OTLN-6l is lower than the spelling accuracy achieved by OTLN. These experimental results show that using one convolution layer is enough to extract useful P300-related separate temporal features for P300 spelling. Using more convolution layers for the extraction of the separate temporal features does not help increasing the spelling accuracy and may cause overfitting which decreases the spelling accuracy.

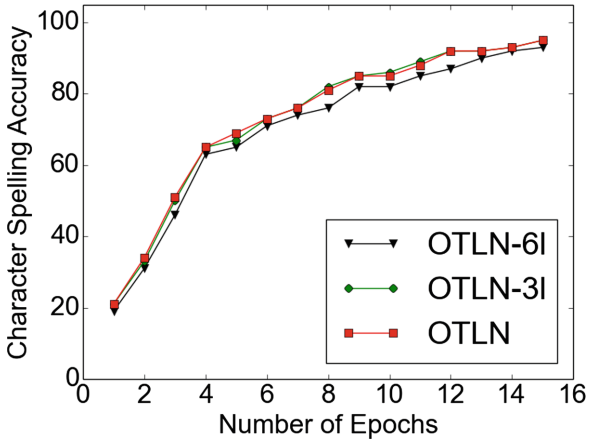

Fig. 4. Spelling accuracy achieved by OTLN, OTLN-3l and OTLN-6l on Dataset III-A.

6.2 Ablation Study on EoCNN

We perform an ablation study on EoCNN. We first remove a CNN from EoCNN. Then, we perform experiments to show the spelling accuracy achieved by the ensemble of the two CNNs left in EoCNN. In this way, we want to show the importance of each separate CNN in EoCNN for character spelling in the P300 speller. The experimental results are shown in Table 10. In this table, "-" indicates that we remove a given CNN from EoCNN. For example, "EoCNN-OSLN"

indicates that we remove OSLN from EoCNN. The experimental results show that after removing any of the individual CNNs from EoCNN, the spelling accuracy achieved by the ensemble of the two CNNs left is lower, compared with the spelling accuracy achieved by EoCNN when none of the individual CNNs is removed. This shows that we need to combine all three CNNs (i.e., OSLN, OTLN, and OCLNN) in EoCNN in order to achieve high spelling accuracy.

Table 10. Spelling accuracy achieved by EoCNN after removing a separate CNN.

Network	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EoCNN-OTLN	**23**	**39**	58	67	75	**81**	82	**86**	86	91	93	96	96	97	**99**
EoCNN-OSLN	22	36	57	66	73	79	80	84	**89**	92	92	95	95	97	98
EoCNN-OCLNN	22	35	55	67	75	79	80	82	83	89	90	93	95	97	98
EoCNN	**23**	**39**	**61**	**68**	**76**	**81**	**84**	**86**	88	**93**	**95**	**98**	**97**	**99**	**99**

6.3 Exploration on the Importance of Extracting P300-Related Temporal Features from Raw Signals

We explore the importance of extracting P300-related temporal features from raw signals. We addressed this issue in Sect. 2. We build two sets of networks, called "RAW_networks" and "unRAW_networks", respectively. RAW_networks contains EoCNN, EoCNN-OSLN, EoCNN-OTLN, EoCNN-OCLNN and OCLNN. All the networks in RAW_networks extract P300-related temporal features from raw signals. unRAW_networks contains CCNN, CNN-R,

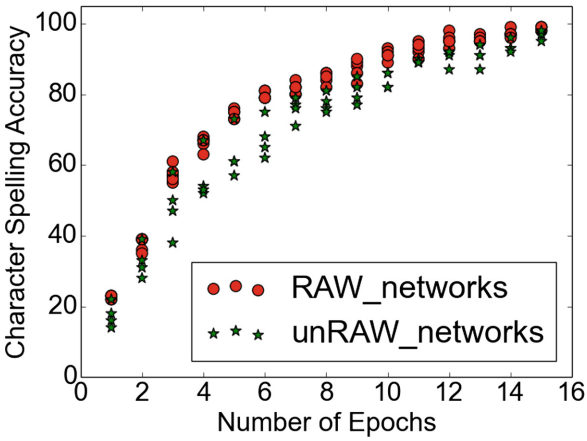

Fig. 5. Spelling accuracy achieved by networks in RAW_networks and networks in unRAW_networks on Dataset III-A.

and BN3. All the networks in unRAW_networks extract P300-related temporal features from abstract temporal signals (the feature maps generated by the spatial convolution layer). We perform experiments to show the spelling accuracy achieved by each network in RAW_networks and the spelling accuracy achieved by each network in unRAW_networks.

The experimental results are shown in Fig. 5. In this figure, the spelling accuracy achieved by the networks in RAW_networks and the spelling accuracy achieved by the networks in unRAW_networks are plotted in different shapes and colors. This figure shows that in most cases, the spelling accuracy achieved by the networks in RAW_networks is higher than the spelling accuracy achieved by the networks in unRAW_networks. This shows that extracting P300-related temporal features from raw signals is able to achieve higher spelling accuracy than extracting P300-related temporal features from abstract signals. These experimental results support our statement in Sect. 2.

7 Conclusions and Future Work

In this paper, we propose a novel and effective network, called EoCNN, for character spelling in the P300 speller. Our EoCNN uses an ensemble of three different CNNs for P300 spelling. These three CNNs extract different useful P300-related features. Experimental results on three datasets show that the spelling accuracy achieved by our network is higher than the spelling accuracy achieved by other methods. Also, the communication speed of the P300 speller based on our network is higher than the communication speed of the P300 speller based on other methods.

The future work includes two aspects. The first aspect is to evaluate the performance of our proposed network via an online P300 speller. The online P300 speller helps the BCI users spell characters in real time. Thus, the performance of the online P300 speller based on our proposed network is able to provide more accurate evaluation for the usage of this BCI system in people's real life. The second aspect of the future work is to evaluate our network with more subjects in terms of spelling accuracy and ITR. Evaluating our network using the EEG signals from more subjects is able to further prove that our network can solve the problem of the subject-to-subject variability in brain signals and achieve high spelling accuracy and ITR across subjects.

References

1. Blankertz, B.: BCI competition II (2003). http://www.bbci.de/competition/ii/
2. Blankertz, B.: BCI competition III (2008). http://www.bbci.de/competition/iii/
3. Bonnet, L., Lotte, F., Lécuyer, A.: Two brains, one game: design and evaluation of a multiuser bci video game based on motor imagery. IEEE Trans. Comput. Intell. AI Games **5**(2), 185–198 (2013)
4. Bostanov, V.: BCI competition 2003-data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. IEEE Trans. Biomed. Eng. **51**(6), 1057–1061 (2004)

5. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) Proceedings of COMPSTAT 2010, pp. 177–186. Physica-Verlag, Heidelberg (2010). https://doi.org/10.1007/978-3-7908-2604-3_16
6. Cecotti, H., Graser, A.: Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE Trans. Pattern Anal. Mach. Intell. **33**(3), 433–445 (2011)
7. Chollet, F., et al.: Keras (2015). https://github.com/keras-team/keras
8. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Ann. Oper. Res. **134**(1), 19–67 (2005)
9. Faux, S.F., Torello, M.W., McCarley, R.W., Shenton, M.E., Duffy, F.H.: P300 in schizophrenia: confirmation and statistical validation of temporal region deficit in P300 topography. Biol. Psychiatry **23**(8), 776–790 (1988)
10. Fazel-Rezai, R., Allison, B.Z., Guger, C., Sellers, E.W., Kleih, S.C., Kübler, A.: P300 brain computer interface: current challenges and emerging trends. Front. Neuroeng. **5**, 14 (2012)
11. Hoffmann, U., Vesin, J.M., Ebrahimi, T.: Spatial filters for the classification of event-related potentials, Technical report (2006)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
13. Lin, C.T., Lin, B.S., et al.: Brain computer interface-based smart living environmental auto-adjustment control system in UPnP home networking. IEEE Syst. J. **8**(2), 363–370 (2014)
14. Lin, C.T., Tsai, S.F., Ko, L.W.: EEG-based learning system for online motion sickness level estimation in a dynamic vehicle environment. IEEE Trans. Neural Netw. Learn. Syst. **24**(10), 1689–1700 (2013)
15. Liu, M., Wu, W., Gu, Z., Yu, Z., Qi, F., Li, Y.: Deep learning based on batch normalization for P300 signal detection. Neurocomputing **275**, 288–297 (2018)
16. Manor, R., Geva, A.B.: Convolutional neural network for multi-category rapid serial visual presentation BCI. Front. Comput. Neurosci. **9**, 146 (2015)
17. Mennes, M., Wouters, H., Vanrumste, B., Lagae, L., Stiers, P.: Validation of ICA as a tool to remove eye movement artifacts from EEG/ERP. Psychophysiology **47**(6), 1142–1150 (2010)
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814 (2010)
19. Pires, G., Nunes, U., Castelo-Branco, M.: Statistical spatial filtering for a P300-based BCI: tests in able-bodied, and patients with cerebral palsy and amyotrophic lateral sclerosis. J. Neurosci. Methods **195**(2), 270–281 (2011)
20. Polich, J.: Updating P300: an integrative theory of P3a and P3b. Clin. Neurophysiol. **118**(10), 2128–2148 (2007)
21. Rakotomamonjy, A., Guigue, V.: BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller. IEEE Trans. Biomed. Eng. **55**(3), 1147–1154 (2008)
22. Rivet, B., Souloumiac, A., et al.: xDAWN algorithm to enhance evoked potentials: application to brain-computer interface. IEEE Trans. Biomed. Eng. **56**(8), 2035–2043 (2009)
23. Shan, H., Liu, Y., Stefanov, T.: A simple convolutional neural network for accurate P300 detection and character spelling in brain computer interface. In: 27th International Joint Conference on Artificial Intelligence (IJCAI), pp. 1604–1610 (2018)

24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
26. Wolpaw, J., Wolpaw, E.W.: Brain-Computer Interfaces: Principles and Practice. OUP, Oxford (2012)
27. Wolpaw, J.R., Ramoser, H., McFarland, D.J., Pfurtscheller, G.: EEG-based communication: improved accuracy by response verification. IEEE Trans. Rehabil. Eng. **6**(3), 326–333 (1998)