Towards high performance and efficient brain computer interface character speller : convolutional neural network based methods

Shan, H.

Author: Shan, H.
Title: Towards high performance and efficient brain computer interface character speller : convolutional neural network based methods
Issue Date: 2020-02-25

Chapter 4

Ensemble of Convolutional Neural Networks for P300 Signal Detection and Character Spelling

Hongchang Shan, Yu Liu, and Todor Stefanov,
"Ensemble of Convolutional Neural Networks for P300 Speller in Brain Computer Interface,"
In *Proceedings of the 28th International Conference on Artificial Neural Networks.*, pp., Munich, Germany, September 17-19, 2019.

IN Chapter 3, we introduce our simple, yet effective OCLNN for the P300 signal detection and character spelling. This CNN solves the problems of the state-of-the-art CNNs for the P300 speller in [CG11, MG15, LWG$^+$18] by performing the spatial convolution and temporal convolution in its first layer using raw signals. Unfortunately, OCLNN still has some limitations to extract some relevant and important features related to P300 signals. OCLNN performs the spatial convolution and the temporal convolution together, thereby realizing a joint spatial-temporal convolution in the first layer. This spatial-temporal convolution extracts only P300-related joint spatial-temporal features in its single convolution layer. OCLNN does not extract P300-related separate temporal features and separate spatial features. These separate temporal features and separate spatial features have proven to be very important for the P300 speller [FTM$^+$88, Pol07, PNCB11, HVE06]. Adding several temporal or spatial convolution layers following the first spatial-temporal convolution layer of OCLNN is a potential method to enable OCLNN to learn P300-related separate spatial or separate temporal features. Nevertheless, such method cannot learn well P300-related separate temporal or spatial features due to the loss of raw information. The raw information loss happens because the input to these additional temporal or spatial convolution layers for OCLNN is the abstract signals generated by the first spatial-temporal convolution layer instead of raw signals.

In order to solve this problem of OCLNN, we proposes a novel network, which combines OCLNN with two other novel CNNs, we have devised, in order to learn well the aforementioned P300-related separate spatial and separate temporal features, which are not extracted by OCLNN, together with the spatial-temporal features extracted by OCLNN. The novel contributions of this chapter are the following:

- Each of the two novel CNNs has only one convolution layer. One of the novel CNNs performs only the temporal convolution in its convolution layer (the first layer) to learn P300-related separate temporal features. The other novel CNN performs only the spatial convolution in its convolution layer (the first layer) to learn P300-related separate spatial features. These two novel CNNs are able to learn well P300-related separate temporal and separate spatial features because the input to each of the two novel CNNs is raw signals. In addition, we propose a novel network which is an ensemble of these two novel CNNs and OCLNN. This network extracts more useful P300-related features than OCLNN alone and is able to achieve higher P300 signal detection accuracy and character spelling accuracy than OCLNN.

- Experimental results on three benchmark datasets show that our proposed ensemble of CNNs is able to increase the P300 signal detection accuracy, the character spelling accuracy, and the communication speed achieved by OCLNN with up to 4.32%, 5%, and 6.05 bits/min, respectively. Also, our proposed ensemble of CNNs outperforms other related methods with a significant P300 signal detection accuracy improvement up to 18.55%, a significant character spelling accuracy improvement up to 38.72%, and a significant communication speed improvement up to 21.75 bits/min. In terms of the network complexity, the complexity of our proposed ensemble of CNNs is lower than the complexity of the CNN in [MG15], and higher than the complexity of OCLNN and the CNNs in [CG11, LWG+18].

The rest of the chapter is organized as follows: Section 4.1 presents our proposed network (ensemble of CNNs) for the P300 speller. Section 4.2 compares the complexity, the P300 signal detection accuacy, the character spelling accuracy, and the communication speed between our network and other related methods for the P300 speller. Section 4.3 analyzes our proposed two novel 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 4.4 ends this chapter with conclusions.

4.1 Proposed Network

This section introduces our proposed network for the P300 signal detection and character spelling in the P300 speller. We call our proposed network Ensemble of Convolutional Neural Networks (EoCNN). EoCNN combines two novel CNNs, we have devised, together with our proposed OCLNN presented in Chapter 3. We call these two novel CNNs as follows: One Spatial Layer Network (OSLN) and One Temporal Layer Network (OTLN).

4.1.1 Ensemble of Convolutional Neural Networks

The workflow of our EoCNN is shown in Figure 4.1. First, the EEG signals are preprocessed to construct the input tensor. For the details of the construction of the input tensor, please refer to Section 3.2.1. Then, the input tensor is sent to three different CNNs, i.e., OSLN, OTLN, and OCLNN. OSLN and OTLN are described in Section 4.1.2. OCLNN is our proposed simple CNN presented in Chapter 3. 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 the P300 signal detection and character spelling in the P300 speller. The detection of P300 signals and the inference of characters by using EoCNN is introduced in Section 4.1.4.

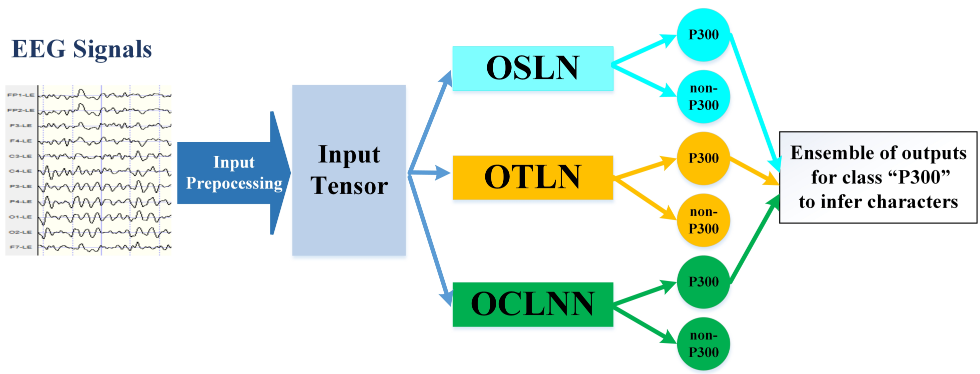

Figure 4.1: Workflow of our EoCNN

4.1.2 Proposed OSLN and OTLN

The architectures of our proposed OSLN and OTLN are described in Table 4.1 and Table 4.2, respectively. OSLN and OTLN are used in EoCNN (see Section 4.1.1), 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.

Table 4.1: OSLN architecture.

Layer	Operation	Kernel	Feature Maps or Neurons
1	Convolution	(1,C)	16
	Dropout	—	—
Output	Fully-Connected	—	2

Table 4.2: OTLN architecture.

Layer	Operation	Kernel	Feature Maps or Neurons
1	Convolution	(N/15, 1)	16
	Dropout	—	—
Output	Fully-Connected	—	2

Layer 1 of OSLN (see Table 4.1) 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 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 only part of all sensors [CG11, MG15, LWG$^+$18, SLS18]. The input to this layer is 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 Output of OSLN.

Layer 1 of OTLN (see Table 4.2) performs the temporal convolution operation with the kernel size (N/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 (N/15, 1) because Chapter 3 has shown that 1/15 of the temporal signal samples is a proper receptive field for a CNN to learn P300-related temporal features. The input to this layer is raw signals, so this layer learns P300-related separate temporal features from raw signals. This layer generates 16 feature maps, which are the input to Layer Output of OTLN.

In both Layer 1 of OSLN and Layer 1 of OTLN, the activation function is the Rectified Linear Unit (ReLU) function. We employ Dropout [SH$^+$14], with a rate of 0.4, to prevent OSLN and OTLN from overfitting (introduced in Section 2.2.3).

Layer Output of OSLN (see Table 4.1) and Layer Output of OTLN (see Table 4.2) are the same. Layer Output 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 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 only 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. However, these abstract signals include only the time domain and do not have the space domain because the first convolution layer uses a receptive field including all C sensors. Thus, these abstract signals cannot be used to extract further spatial features. OTLN also uses only one convolution layer because one convolution layer is enough to extract useful P300-related separate temporal features (as shown and discussed later in Section 4.3.1).

4.1.3 Training

The training process used for our EoCNN is the same as the training process used for our OCLNN (proposed in Chapter 3). For the details of the training process, please refer to Section 3.2.3.

4.1.4 P300 Signal Detection and Character Spelling using EoCNN

We use the outputs of the Softmax function for class "P300" and class "non-P300" in Layer Output of OSLN, OTLN, and OCLNN to detect P300 signals and infer characters. For epoch i and for intensification j we use $P^1_{OS}(i,j)$ to denote the output of the Softmax function for class "P300" in OSLN, $P^0_{OS}(i,j)$ to denote the output of the Softmax function for class "non-P300" in OSLN, $P^1_{OT}(i,j)$ to denote the output of the Softmax function for class "P300" in OTLN, $P^0_{OT}(i,j)$ to denote the output of the Softmax function for class "non-P300" in OTLN, $P^1_{OCL}(i,j)$ to denote the output of the Softmax function for class "P300" in OCLNN, $P^0_{OCL}(i,j)$ to denote the output of the Softmax function for class "non-P300" in OCLNN. Therefore, for epoch i and for intensification j, $P^1_{OS}(i,j)$ denotes the predicted probability by OSLN for class “P300”; $P^0_{OS}(i,j)$ denotes the predicted probability by OSLN for class “non-P300”; $P^1_{OT}(i,j)$ denotes the predicted probability by OTLN for class “P300”; $P^0_{OT}(i,j)$ denotes the predicted probability by OTLN for class “non-P300”; $P^1_{OCL}(i,j)$ denotes the predicted probability by OCLNN for class “P300”; and $P^0_{OCL}(i,j)$ denotes the predicted probability by OCLNN for class “non-P300”.

We use Equation (4.1), (4.2), and (4.3) for the detection of P300 signals, where $P^1_{EoC}(i,j)$ denotes the predicted probability by EoCNN for class “P300”; $P^0_{EoC}(i,j)$ denotes the predicted probability by EoCNN for class “non-P300”; EoC denotes our EoCNN classifier; and $X_{(i,j)}$ denotes the input tensor to be classified. Equation (4.1) and (4.2) show the ensemble processing of the outputs from OSLN, OTLN, and OCLNN. Equation (4.3) shows the detection of a P300 signal. In this equation, $EoC(X_{(i,j)}) = 1$ means that EoCNN detects a P300 signal from the input tensor $X_{(i,j)}$ and $EoC(X_{(i,j)}) = 0$ means that EoCNN does not detect a P300 signal from this input tensor. After using Equation (4.3) to detect P300 signals, we can assess the performance of our proposed EoCNN in terms of the P300 signal detection accuracy (see Section 4.2.2).

$$P^1_{EoC}(i,j) = \frac{1}{3} \times (P^1_{OS}(i,j) + P^1_{OT}(i,j) + P^1_{OCL}(i,j)) \tag{4.1}$$

$$P^0_{EoC}(i,j) = \frac{1}{3} \times (P^0_{OS}(i,j) + P^0_{OT}(i,j) + P^0_{OCL}(i,j)) \tag{4.2}$$

$$EoC(X_{(i,j)}) = \begin{cases} 1 & if \quad P^1_{EoC}(i,j) > P^0_{EoC}(i,j) \\ 0 & otherwise \end{cases} \tag{4.3}$$

We use $P^1_{EoC}(i,j)$, the output of EoCNN for class “P300”, to calculate the position of the target character in the P300 speller character matrix. For the detailed calculation process, please refer to Section 2.4.2, Equation (2.30), (2.31), and (2.32).

4.2 Experimental Evaluation

The experimental setup used for the evaluation, presented in this section, is the same as the experimental setup used in Chapter 3 (for detailed description of the experimental setup please see Section 3.3.1). We first compare the complexity of our EoCNN with the complexity of other related CNNs for the P300 speller in Section 4.2.1. Then, we compare the P300 signal detection accuracy achieved by our EoCNN and other related methods in Section 4.2.2. Also, we compare the character spelling accuracy achieved by our EoCNN and other related methods in Section 4.2.3. Finally, we compare the ITR of the P300 speller based on our EoCNN and other related methods in Section 4.2.4.

4.2.1 Complexity

In this section, we compare the complexity of EoCNN, in terms of the number of parameters (explained in Section 3.3.2) and layers, with the complexity of the networks OCLNN (proposed and presented in Chapter 3), CCNN [CG11], BN3 [LWG+18], and CNN-R [MG15] briefly described in Section 3.1. Concerning the complexity of EoCNN, since EoCNN is the ensemble of OSLN, OTLN, and OCLNN, the number of the parameters of EoCNN is calculated as the sum of the number of parameters of OSLN, OTLN, and OCLNN, and the number of the layers used in EoCNN is calculated as the sum of the number of the layers used in OSLN, OTLN, and OCLNN. The complexity of different CNNs is shown in Table 4.3. The first row in the table lists the CNNs, we compare. The second row provides the number of parameters for each CNN. The third row shows the number of layers used in each CNN. Table 4.3 shows that the complexity of EoCNN, in terms of the number of parameters and layers, is higher than the complexity of OCLNN, CCNN, BN3 and lower than the complexity of CNN-R.

Table 4.3: Complexity of different CNNs.

	EoCNN	OCLNN	CCNN	BN3	CNN-R
Parameters	56598	16882	37502	39489	21950818
Layers	6	2	4	5	6

4.2.2 P300 Signal Detection Accuracy

This section compares the P300 signal detection accuracies achieved by EoCNN with the P300 signal detection accuracies achieved by OCLNN (proposed and presented in Chapter 3), CCNN [CG11], BN3 [LWG+18], and CNN-R [MG15] on Dataset II, III-A and III-B.

The P300 signal detection accuracies are shown in Table 4.4. The first row in the table lists the CNNs used for comparison. The second, third, and last row show the P300 signal detection accuracy of the different CNNs on Dataset II, III-A, and III-B, respectively. The numbers are given in percentage (%) and calculated using Equation (3.5). An accuracy number in bold indicates the highest accuracy along a row. "–" in the table means that the accuracy is not reported in the reference paper describing the corresponding CNN.

Table 4.4: P300 signal detection accuracy of different CNNs on Dataset II, III-A, and III-B.

	EoCNN	OCLNN	CCNN	BN3	CNN-R
P300 Accuracy on II	**94.76**	92.41	–	84.44	86.29
P300 Accuracy on III-A	**88.92**	84.60	70.37	75.13	73.06
P300 Accuracy on III-B	**89.65**	86.40	78.19	79.02	79.80

Overall, EoCNN achieves the highest accuracy among all CNNs on Dataset II, III-A, and III-B. It increases the P300 signal detection accuracies achieved by the other CNNs with up to 18.55%. Compared with OCLNN, EoCNN is able to increase the P300 signal detection accuracy achieved by OCLNN with 2.35%, 4.32%, and 3.25% on Dataset II, Dataset III-A, and Dataset III-B, respectively. Compared with the other related CNNs, on Dataset II, EoCNN is able to increase the P300 signal detection accuracy achieved by BN3 and CNN-R with 10.32% and 8.47%, respectively. On Dataset III-A, EoCNN is able to increase the P300 signal detection accuracy achieved by CCNN, BN3, and CNN-R with 18.55%, 13.79%, and 15.86%, respectively. On Dataset III-B, EoCNN is able to increase the P300 signal detection accuracy achieved by CCNN, BN3, and CNN-R with 11.46%, 10.63%, and 9.85%, respectively.

4.2.3 Character Spelling Accuracy

This section compares the character spelling accuracies achieved by our EoCNN and the accuracies achieved by OCLNN, CCNN, BN3, CNN-R, and ESVM [RG08] for Dataset III-A and III-B, as well as the character spelling accuracies achieved by EoCNN and the accuracies achieved by OCLNN, CCNN, BN3, CNN-R, and Bostanov [Bos04] for Dataset II.

The character spelling accuracy achieved by our EoCNN and other methods on Dataset II, Dataset III-A, and Dataset III-B is shown in Table 4.5, 4.6, and 4.7, 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 the accuracy number. The accuracy numbers in these tables are given in percentage (%). 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 4.5: 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 4.6: 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 4.5, 4.6, and 4.7 show that, when compared with OCLNN (proposed and presented in Chapter 3), the spelling accuracy achieved by EoCNN is higher than the spelling accuracy achieved by OCLNN in most cases. EoCNN is able to increase the character spelling accuracy achieved by OCLNN with up to 6.45%, 5%, 5% for Dataset II, Dataset III-A, and Dataset III-B, respectively. However, on epoch number $k = 9$ in Dataset III-A and on epoch number $k = 8$ in Dataset III-B, EoCNN decreases the spelling accuracy achieved by OCLNN. The reason for this is that EoCNN puts equal importance on OSLN, OTLN, and OCLNN in the ensemble processing of the

Table 4.7: 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

outputs from OSLN, OTLN, and OCLNN. For more details of the explaination on this reason please refer to Chapter 6.

Table 4.5, 4.6, and 4.7 also show that, when compared with other related methods, for Dataset II, our EoCNN can increase the spelling accuracy achieved by CCNN, CNN-R, BN3, and Bostanov with up to 38.72%, 12.90%, 19.36%, and 19.35%, respectively. For Dataset III-A, our EoCNN can increase the spelling accuracy achieved by CCNN, CNN-R, BN3, and ESVM with up to 16%, 23%, 7%, and 10%, respectively. For Dataset III-B, our EoCNN can increase the accuracy achieved by CCNN, CNN-R, BN3, and ESVM with up to 16%, 20%, 8%, and 16%, respectively.

Moreover, our method is robust across different subjects. Table 4.5, 4.6, and 4.7 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 4.5, 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 the spelling accuracy for the P300 speller. Such observation is also discussed in more details in [CG11]. The other insight is from the fact that in Dataset II, we need only 2 epochs to achieve a spelling accuracy which is higher than 90% while in Dataset III-A and Dataset III-B, in order to achieve a spelling accuracy higher than 90%, we need at least 10 epochs and 6 epochs, respectively. This indicates 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.

4.2.4 Information Transfer Rate

This section compares the Information Transfer Rate (ITR) of the P300 speller based on our EoCNN and other methods. ITR is calculated using Equation (2.34) and (2.35) (introduced in Section 2.4.3). 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 Table 4.8, 4.9, and 4.10, 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 number is the highest ITR along a row. "–" 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.

Table 4.8: 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	**48.33**	40.25	35.25	28.46	23.86	20.54	18.03	16.07	14.5	13.2	12.12	11.2	10.41	9.72	9.12
CCNN	**26.58**	16.65	22.09	24.73	20.74	17.85	15.67	14.92	13.45	13.2	12.12	11.2	10.41	9.72	9.12
CNN-R	**36.68**	33.18	30.64	26.41	23.86	20.54	18.03	16.07	14.5	13.2	12.12	11.2	10.41	9.72	9.12
BN3	**42.28**	27.06	23.65	20.4	20.74	19.07	16.74	14.92	14.5	13.2	12.12	11.2	10.41	9.72	9.12
OCLNN	**42.28**	37.74	35.25	28.46	23.86	20.54	18.03	16.07	14.5	13.2	12.12	11.2	10.41	9.72	9.12
Bostanov	31.46	**33.18**	30.64	26.41	22.15	20.54	18.03	16.07	14.5	13.2	12.12	11.2	10.41	9.72	9.12

Table 4.9: 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	5.77	9.64	**15.03**	14.44	14.51	13.88	12.96	12.02	11.29	11.35	10.84	10.67	9.71	9.48	8.89
CCNN	2.96	7.33	9.91	9.41	10.18	9.7	**11.21**	10.2	10.63	9.87	9.82	9.25	8.6	8.36	8.51
CNN-R	2.28	5.56	7.03	9.7	9.13	8.99	**9.82**	9.56	9.01	9.11	9.62	8.55	7.94	8.2	8.17
BN3	5.33	9.64	13.87	**14.1**	13.59	12.22	11.69	10.86	10	9.87	9.62	9.44	9.13	8.88	8.69
OCLNN	5.77	9.64	13.11	12.78	**13.59**	13.32	12.44	11.78	11.74	10.91	10.63	10.02	9.32	8.88	8.89
ESVM	2.96	6.96	11.65	11.82	**13.28**	–	–	–	–	9.29	–	–	9.13	–	8.51

We compare the max-ITR[1] achieved by our EoCNN and other methods for the P300 speller. Overall, the max-ITR achieved by our EoCNN is higher than the max-ITR achieved by all other methods. Our EoCNN is able to increase the max-ITR achieved by other methods with up to 21.75 bits/min.

[1]The notion of max-ITR is introduced in Section 3.3.5.

Table 4.10: 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	21.61	**22.4**	20.52	19.23	17.15	16.64	14.9	13.55	12.97	12.31	11.55	10.67	9.92	9.27	8.89
CCNN	11.76	15.3	14.25	14.44	**15.47**	13.88	12.44	12.76	12.22	10.91	10.01	9.07	8.6	8.2	7.69
CNN-R	12.32	12.58	**17.05**	15.14	14.83	13.6	13.49	12.02	11.29	10.91	10.63	10.02	9.32	8.88	8.33
BN3	**18.97**	18.72	18.75	16.2	14.51	14.17	12.96	13.28	12.71	11.81	10.84	10.02	9.13	8.53	8.17
OCLNN	18.32	**20.26**	19.62	18.45	17.15	15.68	14.32	13.82	12.71	12.06	11.3	10.44	9.71	9.27	8.69
ESVM	11.76	**15.78**	15.43	14.44	14.2	–	–	–	–	10.91	–	–	9.51	–	8.33

Table 4.8, 4.9, and 4.10 show that, when compared with OCLNN (proposed and presented in Chapter 3), the max-ITR achieved by our EoCNN is higher than the max-ITR achieved by our OCLNN on all three datasets. Our EoCNN increases the max-ITR achieved by our OCLNN with 6.05 bits/min, 1.44 bits/min, and 2.14 bits/min on Dataset II, III-A, and III-B, respectively.

Table 4.8, 4.9, and 4.10 also show that, when compared with other related methods for the P300 speller, for Dataset II, the max-ITR achieved by our EoCNN is higher than the max-ITR achieved by all other methods, i.e., CCNN, CNN-R, BN3, and Bostanov. Our EoCNN increases the max-ITR achieved by CCNN, CNN-R, BN3, and Bostanov with 21.75 bits/min, 11.65 bits/min, 6.05 bits/min, and 15.15 bits/min, respectively. For Dataset III-A, the max-ITR achieved by our EoCNN is higher than the max-ITR achieved by all other methods, i.e., CCNN, CNN-R, BN3, and ESVM. Our EoCNN increases the max-ITR achieved by CCNN, CNN-R, BN3, and ESVM with 3.82 bits/min, 5.21 bits/min, 0.93 bits/min, and 1.75 bits/min, respectively. For Dataset III-B, the max-ITR achieved by our EoCNN is higher than the max-ITR achieved by all other methods, i.e., CCNN, CNN-R, BN3, and ESVM. Our EoCNN increases the max-ITR achieved by CCNN, CNN-R, BN3, and ESVM with 6.93 bits/min, 5.35 bits/min, 3.43 bits/min, and 6.62 bits/min, respectively.

Our EoCNN increases the max-ITR achieved by our OCLNN, thereby bringing the max-ITR more closer to the theoretically achievable maximum ITR (introduced in Section 2.4.3). Unfortunately, the complexity, in terms of the number of parameters, of EoCNN is 3.35 times higher than the complexity of OCLNN (see Table 4.3). As described in Section 1.1.2, the low complexity of a CNN-based method for P300 character spelling is an important requirement to build efficient P300 spellers that can be used in people's daily life. Therefore, increasing further the complexity of our EoCNN-based P300 speller in order to further increase the max-ITR of the speller is not a suitable way to go in terms of efficiency. Thus, further research efforts are needed to find alternative ways to further increase the max-ITR without sacrificing the efficiency of the EoCNN-based P300 speller. For example, one possible alternative ways is to devise a better character matrix (Figure 2.10) which enables the reduction of the time periods t_1, t_2, and t_3 in the P300 speller experiment (see Section 2.4.2 and Equation (2.34) and (2.35)) . However, such psychology-related research direction is out of the scope of this thesis.

4.3 Discussions

In this section, first, we analyse our proposed OTLN and OSLN in terms of character spelling accuracy and discuss the influence of the number of convolution layers on extracting useful P300-related separate temporal features in Section 4.3.1. Then, we perform an ablation study on EoCNN to show that we need to combine all three CNNs (i.e., OSLN, OTLN, OCLNN) in EoCNN in order to achieve high spelling accuracy in Section 4.3.2. Finally, we explore the importance of extracting P300-related features from raw signals in Section 4.3.3.

In this section, all the experiments are performed by using the experimental setup described in Section 3.3.1. We draw similar conclusions from the experimental results of all datasets, i.e., Dataset III-A, Dataset III-B, and Dataset II. Thus, the experimental results are shown using only Dataset III-A in order to present our conclusions.

4.3.1 Analysis of Our Proposed OTLN and OSLN

First, we perform experiments to show the character spelling accuracy achieved by OTLN and OSLN, respectively. The experimental results are shown in Table 4.11. In this table, the 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 numbers in this table are given in percentage (%). Table 4.11 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. 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 Figure 4.2. 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 separate temporal features does not help increasing the spelling accuracy and may cause overfitting which decreases the spelling accuracy.

Table 4.11: 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**

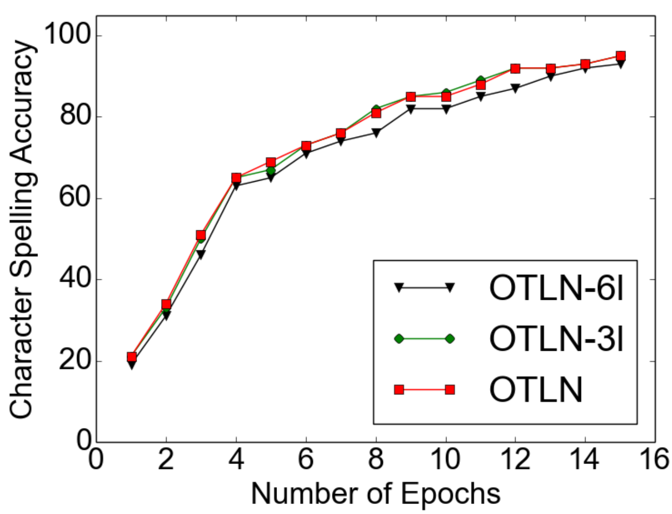

Figure 4.2: Spelling accuracy achieved by OTLN, OTLN-3l and OTLN-6l on Dataset III-A.

4.3.2 Ablation Study on EoCNN

We perform an ablation study on EoCNN to show that we need to combine all three CNNs (i.e., OSLN, OTLN, OCLNN) in EoCNN in order to achieve high spelling accuracy. 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 4.12. 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, OCLNN) in EoCNN in order to achieve high spelling accuracy. The experimental results from Table 4.12 also give us some insights. For example, in most cases, the spelling accuracy achieved by EoCNN-OTLN is higher than the spelling accuracy achieved by EoCNN-OSLN. This shows that P300-related spatial features are more important than P300-related temporal features on increasing the spelling accuracy. This is because a large number of sensors (i.e., 64 sensors) are used to acquire EEG signals in the P300 speller. When using a large number of sensors for the acquisition of EEG signals, we need to put more importance on extracting P300-related spatial features in order to achieve high spelling accuracy (For more explanation, please see Chapter 6).

Table 4.12: 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**

4.3.3 Exploration on the Importance of Extracting P300-related Features from Raw Signals

We explore the importance of extracting P300-related temporal features from raw signals. We consider two sets of networks. These two sets of networks are called "RAW_networks" and "unRAW_networks", respectively. RAW_networks include networks EoCNN, OCLNN, EoCNN-OSLN, EoCNN-OTLN, and EoCNN-OCLNN. All the networks in set RAW_networks extract P300-related temporal features from only raw signals. unRAW_networks include networks CCNN, CNN-R and BN3. All the networks in set unRAW_networks extract P300-related temporal features from abstract signals. We perform experiments to show the spelling accuracy achieved by each network in set RAW_networks and the spelling accuracy achieved by each network in set unRAW_networks.

The experimental results are shown in Figure 4.3. In this figure, the spelling accuracy achieved by the networks in set RAW_networks and the spelling accuracy achieved by the networks in set unRAW_networks are plotted in different shapes and colors. This figure shows that in most cases, the spelling accuracy achieved by the networks in set RAW_networks is higher than the spelling accuracy achieved by the networks in set unRAW_networks. This fact indicates 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.

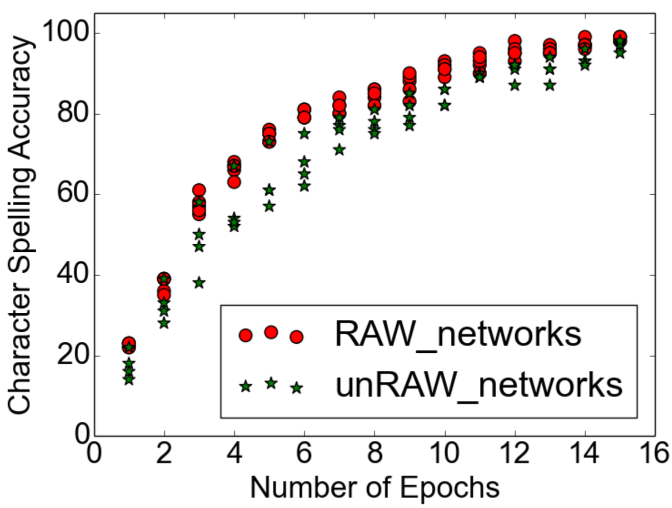

Figure 4.3: Spelling accuracy achieved by networks in set RAW_networks and networks in set unRAW_networks on Dataset III-A.

4.4 Conclusions

In this chapter, we propose a novel and effective network, called EoCNN, for the P300 signal detection and 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 our EoCNN increases the P300 signal detection accuracy, the character spelling accuracy, and the ITR achieved by OCLNN (proposed and presented in Chapter 3) and other related methods for the P300 speller. In addition, our EoCNN is robust across different subjects.

Unfortunately, the complexity of our EoCNN is only lower than the complexity of CNN-R, and higher than the complexity of OCLNN, CCNN, and BN3. Thus, when compared to CNN-R, we should use our EoCNN for the P300 speller because our EoCNN has lower complexity and achieves higher P300 signal detection accuracy, character spelling accuracy, and ITR than CNN-R. When compared with OCLNN, CCNN, and BN3, if the hardware platform used in an efficient P300-based BCI system cannot support the high complexity of EoCNN, we need to choose a network among OCLNN, CCNN, and BN3 to be used for the P300 speller. In this case, we should use OCLNN because OCLNN is better than CCNN and BN3 for the P300 speller (For detailed explanation on why OCLNN is better than CCNN and BN3 for the P300 speller, please see Section 3.3.5). If the hardware platform used in an efficient P300-based BCI system can support the complexity of EoCNN, we should use EoCNN in such P300-based BCI system because EoCNN is able to achieve higher P300 signal detection accuracy, character spelling accuracy, and ITR than OCLNN, CCNN, and BN3 for the P300 speller.