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 5

A Novel Sensor Selection Method based on Convolutional Neural Network for P300 Speller

Hongchang Shan, and Todor Stefanov,
"SLES: A Novel CNN-based Method for Sensor Reduction in P300 Speller,"
In *Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'19)*, Berlin, Germany, July 23-27, 2019.

Hongchang Shan, and Todor Stefanov,
"A Novel Sensor Selection Method based on Convolutional Neural Network for P300 Speller in Brain Computer Interface",
The 56th ACM/IEEE Design Automation Conference (DAC'19) WIP session, Las Vegas, NV, USA, June 2-6, 2019.

P300 spellers are still not used in human's daily life and remain in an experimental stage at research labs. Some of the reasons for this situation are : 1) Current popular EEG headsets in BCI systems, used for P300 spellers, utilize a large number of sensors to achieve high spelling accuracy. For example, the BCI systems Brain Products ActiCHamp [Act], g.HIamp [g.H], and Biosemi ActiveTwo [Bio18] utilize up to 160, 256, and 280 sensors, respectively. The price of the EEG headset is significantly high when the number of sensors is large because a lot of sensors require a complicated electrode cap and a lot of amplifier channels. For example, a 280-sensor BCI system (e.g., BioSemi ActiveTwo) costs around 87000 dollars while a 14-sensor BCI system (e.g., EMOTIV EPOC+ [EMO]) costs 799 dollars; 2) Utilizing a large number of sensors makes the P300 speller to consume a lot of power, which is unacceptable for a battery-powered mobile BCI practical system. Such system utilizes a wireless EEG headset and a resource-constrained hardware platform for data processing. A large number of sensors increases the amount of the data needed to be recorded and processed, thereby increasing the power consumption of the wireless BCI headset and the hardware platform. This does not allow a mobile practical P300 speller to work for a long time period on a single battery charge; 3) Utilizing a large number of sensors strengthens the user's discomfort and increases the installation time of the P300 speller.

To address the aforementioned problems caused by the utilization of a large number of sensors, sensor selection methods could be used to select an appropriate sensor subset from an initial large set of sensors while keeping acceptable spelling accuracy. So, a good sensor selection method should enable substantial reduction of the sensors needed to acquire brain signals. Therefore, good sensor selection methods are in urgent need for designing comfortable, cheap, and power-efficient P300 spellers and for promoting such P300 spellers into the human's daily life. Sensor selection methods for the P300 speller have been studied in recent years. For example, [RG08] [RSG+09] [CRC+10] [CR+11] utilize a backward elimination algorithm as a sensor selection strategy. These works propose different ranking functions to evaluate and eliminate sensors such as the P300 signal detection accuracy, the P300 spelling accuracy [CR+11], the C_{cs} score [RG08], Signal to Signal and Noise Ratio (SSNR) [RSG+09] [CRC+10] [CR+11], Area Under the Receiver Operating Characteristic (AUC) [CRT+14]. Alternatively, [CG11] and [LWG+18] directly select the important sensors for a given user by analysing the weights of a trained neural network. Unfortunately, the aforementioned sensor selection methods cannot select an appropriate sensor subset such that they can further reduce the number of sensors used to acquire brain signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. As a consequence, the cost, power consumption, and discomfort of a P300 speller are still unacceptably high when using the aforementioned sensor selection methods to design and configure P300 spellers. In order to further reduce the cost and power consumption of a P300 speller, we propose an effective sensor selection method based on a specific novel CNN, i.e., the OSLN, we have devised and presented in Chapter 4. The novel contributions of this chapter are the following:

- We parameterize the OSLN with the number of sensors used for the acquisition of EEG signals. Our sensor selection method uses this parameterized CNN to evaluate and rank the sensors during the sensor selection process. This method features an iterative parameterized backward elimination algorithm to eliminate and select sensors. The parameter, configured in this backward elimination algorithm, controls the training frequency of the CNN and the number of sensors to eliminate in every iteration.

- We perform experiments on three benchmark datasets and compare the minimal number of sensors selected by our proposed method and other selection methods needed to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. The results show that, compared with the minimal number of sensors selected by other methods, our method can reduce this number with up to 44 sensors.

The rest of the chapter is organized as follows. Section 5.1 describes the related work. Section 5.2 presents our proposed sensor selection method. Section 5.3 describes the experimental setup and the experimental results on the comparison of the minimal number of sensors selected by our proposed method and other sensor selection methods to acquire brain signals for the P300 speller. Section 5.4 discusses how the number of sensors eliminated in an iteration influences the performance of our proposed method as well as how the CNN network architecture influences the sensor selection process. Section 5.5 ends the chapter with conclusions.

5.1 Related Work

In this section, we describe the related works on sensor selection methods for the P300 speller in BCI.

[RG08] [CR+11] utilize a backward elimination algorithm as a sensor selection strategy. Different ranking functions are proposed to evaluate and eliminate sensors. These ranking functions include the P300 detection accuracy, the average spelling accuracy across different epochs [CR+11], the C_{cs} score [RG08], Signal to Signal and Noise Ratio (SSNR) [CR+11], and Area Under the Receiver Operating Characteristic (AUC) [CRT+14]. In order to select a sensor subset, the backward elimination algorithm either eliminates one sensor [CR+11] or a group of sensors [RG08] in each iteration of the algorithm. Starting with a set of n sensors in an iteration, the backward elimination algorithm removes each sensor in the current sensor set and evaluates the resulting subsets with $(n - 1)$ sensors using the aforementioned ranking functions. The sensor or the group of sensors which removal maximizes the ranking score is eliminated. In contrast to these methods, we proposes a novel ranking function (see Section 5.2.3) based on the OSLN we have devised and presented in Chapter 4. Experimental results (see Section 5.3.2) show that our sensor selection method is able to select a sensor subset with smaller number of sensors needed to acquire the EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used, compared with the sensor subset selected by the aforementioned sensor selection methods. Therefore, our sensor selection method can further reduce the cost and power consumption of the P300 speller.

[CG11] and [LWG$^+$18] propose CNN-based classifiers for character spelling in the P300 speller. By analysing the weights of the spatial convolution layer of their trained CNNs, they determine which sensors are more important in the sensor set. This can be a potential sensor selection method for the P300 speller. However, the problem of such potential method is that it loses important information needed for proper sensor selection. The aforementioned CNNs have multiple convolution layers. The information needed for proper sensor selection is distributed over the weights of all convolution layers. In [CG11] and [LWG$^+$18], only the weights of the first layer are used for analysis and sensor selection because the weights of the other convolution layers can hardly be used for sensor selection (for the detailed explanation of the reason for this please refer to Section 5.4.2). Thus, the aforementioned methods cannot use all the information available for proper sensor selection. In contrast to the aforementioned CNNs, our proposed OSLN has only one convolution layer and this layer performs the spatial convolution operation. All the information needed for sensor selection is captured by the weights of this single spatial convolution layer. Moreover, our CNN has similar ability to extract very useful P300-related features compared to the aforementioned CNNs (see Table 4.11 and Table 3.8). We analyse the weights of the single spatial convolution layer in our CNN to select sensors. Thus, our method uses all the information available for proper sensor selection compared to the aforementioned methods. As a result, our method can select more appropriate sensor subsets and further reduce the minimal number of sensors needed to acquire brain signals without losing spelling accuracy. For more detailed discussion see Section 5.4.2.

5.2 Our Sensor Selection Method

In this section, we present our novel iterative sensor selection method for the P300 speller. We call it Spatial Learning based Elimination Selection (SLES).

5.2.1 Spatial Learning based Elimination Selection

Our SLES method is described in Algorithm 1. The symbols used in Algorithm 1 and their corresponding descriptions are listed in Table 5.1. The input of SLES is the initial sensor set S and the parameter E_s. The output of SLES is a set of selected sensor subsets SUB. For each iteration in Algorithm 1, SLES trains $OSLN_{(S)}$ with the input signals recorded with the sensors in sensor set S (see Line 2 in Algorithm 1). $OSLN_{(S)}$ (described in Section 5.2.2) is the parameterized version of the OSLN (proposed in Chapter 4) with S as a parameter. After training $OSLN_{(S)}$, the ranking scores $score_j$ for all sensors s_j in sensor set S are calculated (Line 3-4) using

$OSLN_{(S)}$ and Equation (5.2) explained in Section 5.2.3. The sensor with the minimal score is found and removed from sensor set S (Lines 6-7). This reduced sensor set S is the selected sensor subset in this iteration (Line 8). The input parameter E_s controls the training frequency of $OSLN_{(S)}$ (Line 1) and the number of sensors to eliminate after training $OSLN_{(S)}$ (Line 5).

Table 5.1: The symbols used in Algorithm 1.

Symbol	Description
S	Sensor set.
s_j	The jth sensor in S.
C	Number of sensors in the initial sensor set.
SUB	A set of selected sensor subsets.
sub_m	A selected sensor subset with m sensors.
$OSLN_{(S)}$	The novel parameterized CNN given in Section 5.2.2
E_s	Number of sensors to eliminate in an iteration.
$score_j$	The ranking score for s_j.
s_{remove}	The sensor to remove.

5.2.2 Parameterized OSLN

In this section, we describe in details the $OSLN_{(S)}$ (used in Algorithm 1), which is the parameterized version of the OSLN, proposed and presented in Chapter 4.

5.2.2.1 Input Tensor

The input to $OSLN_{(S)}$ is the tensor ($N \times |S|$) shown in Figure 5.1. S is the sensor set used in Algorithm 1. x_{ji} denotes the ith temporal signal sample in the time domain and this signal sample is recorded with sensor s_j in sensor set S in the space domain. $OSLN_{(S)}$ is parameterized by S because the input tensor to $OSLN_{(S)}$ is constructed by the EEG signal samples acquired using the sensors in sensor set S and S is changed in each main iteration of Algorithm 1 (see Line 7). N denotes the number of temporal

Algorithm 1: Proposed SLES algorithm.

Input: Set S = $\{s_1, s_2, ..., s_j, ...\ s_C\}$, E_s;
Output: Set SUB = $\{sub_1, sub_2, ...\ , sub_m, ...\ sub_{C-1}\}$;
for $1 \le k \le C/E_s$ **do**
  Train a $OSLN_{(S)}$ with the input signals recorded using S;
  for $s_j \in S$ **do**
    Calculate $score_j$ using $OSLN_{(S)}$ and Equation (5.2);
  for $1 \le m \le E_s$ **do**
    s_{remove} = $\underset{s_j \in S}{argmin}\ \{score_j\}$;
    $S \leftarrow S - s_{remove}$;
    $sub_{(C-E_s*(k-1)-m)} \leftarrow S$;

signal samples. These temporal signal samples are preprocessed in the same way as explained in Section 3.2.1 of Chapter 3.

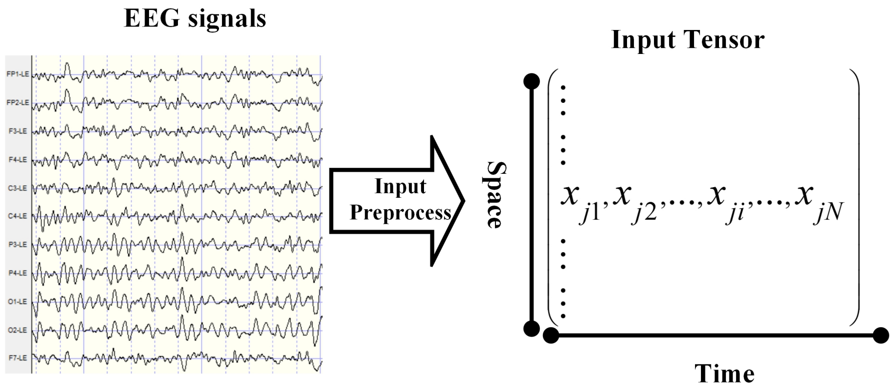

Figure 5.1: Input tensor to $OSLN_{(S)}$, where $s_j \in S$.

5.2.2.2 Network Architecture

Table 5.2 shows the details of the $OSLN_{(S)}$ architecture. The first column shows the name of the layers. The second column shows the operation performed in the corresponding layer. The third column shows the kernel size in the convolution layer. The fourth column shows how many feature maps or neurons are utilized in the convolution or fully-connected layer. The difference between $OSLN_{(S)}$ and OSLN (see Table 4.1 in Section 4.1.2) is Layer 1, i.e., the convolution layer. Thus, we describe only Layer 1 of $OSLN_{(S)}$ in this section.

Table 5.2: $OSLN_{(S)}$ architecture.

Layer	Operation	Kernel Size	Feature Maps/Neurons
1	Convolution	(1,$\|S\|$)	16
	Dropout	—	—
2	Fully-Connected	—	2

In Layer 1, $OSLN_{(S)}$ performs a spatial convolution operation to extract the spatial features related to P300 signals from the input tensor. The detailed calculation in this convolution operation is shown in Equation (5.1), where f_{ki} denotes the ith datum in the kth feature map. w_{kj} denotes the jth weight of the filter and this filter outputs abstract data for the kth feature map. The activation function we utilize in this layer is the Rectified Linear Unit (ReLU). In this layer, we utilize Dropout in order to prevent the network from overfitting. In this layer, we do not use bias in the convolution operation, thus all the learned features are captured by the weights w_{kj}. This layer outputs 16 feature maps in total.

$$f_{ki} = \sum_{s_j \in S} x_{ji} w_{kj} \tag{5.1}$$

5.2.3 Ranking Function

Our proposed novel ranking function used in SLES is given in Equation (5.2), where $score_j$ is the ranking score for sensor s_j used in Algorithm 1. w_{kj} are the weights described in Equation (5.1). These weights are obtained from the trained $OSLN_{(S)}$, described in Section 5.2.2.2 and used in Algorithm 1. Note that we take the absolute value of the weights in Equation (5.2) because weights with a large negative value also indicate that the corresponding sensors are important in sensor set S. We use the absolute values of the weights from the spatial convolution layer (i.e., Layer 1 in Table 5.2) of the trained $OSLN_{(S)}$ in the ranking function to rank the sensors in the sensor set because [CG11] and [LWG$^+$18] have shown that analyzing the weights of the spatial convolution layer from trained CNNs for the P300 speller is a potential method to determine which sensors are more important in the sensor set. For details, please refer to [CG11] and [LWG$^+$18].

$$score_j = \sum_{k=1}^{16} |w_{kj}| \tag{5.2}$$

We have proposed three CNNs with one convolution layer, i.e., OSLN, OCLNN (proposed and presented in Chapter 3), and OTLN (proposed and presented in Section 4.1.2 of Chapter 4). Our ranking function to rank the sensors in the sensor set is based on the parameterized OSLN instead of a parameterized version of OCLNN as well as instead of a parameterized version of OTLN. The reason for this is explained in Section 5.4.2.

5.3 Experimental Evaluation

In this section, we present the experiments, we have performed, in order to compare the minimal number of sensors selected by our method and other methods to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. We first introduce our experimental setup and then we present and analyse the obtained experimental results.

5.3.1 Experimental Setup

To perform the experiments, we use 3 different implementations of the P300 speller: the OCLNN-based P300 speller (proposed and presented in Chapter 3), the EoCNN-based P300 speller (proposed and presented in Chapter 4), and the SVM-based P300 speller [RG08]. We want to confirm the robustness of our SLES method by showing that our method is effective for different P300 speller implementations.

We compare our SLES method with 12 other sensor selection methods. These methods are summarized in Table 5.3. In this table, the first row gives the names of the different methods. The second row describes the sensor elimination algorithms used in the methods, where BE-1 denotes a backward elimination algorithm which eliminates one sensor at a time; BE-4 denotes a backward elimination algorithm which eliminates 4 sensors at a time; "–" denotes that the corresponding method does not use a backward elimination algorithm. The last row indicates the ranking functions used in the methods, where P300 denotes the P300 detection accuracy; Char denotes the average character spelling accuracy across all epochs; AUC denotes Area Under the Receiver Operating Characteristic [CRT+14]; C_{cs} denotes the ranking score proposed in [RG08]; SSNR denotes Signal to Signal and Noise Ratio [CR+11]; CCNN and BN3 denote that the corresponding method selects sensors by analysing the weights obtained from the trained networks CCNN [CG11] and BN3 [LWG+18], respectively.

Table 5.3: Methods compared with SLES.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Algo.	BE-1	BE-1	BE-1	BE-1	BE-1	BE-4	BE-4	BE-4	BE-4	BE-4	–	–
Function	P300	Char	AUC	C_{cs}	SSNR	P300	Char	AUC	C_{cs}	SSNR	CCNN	BN3

We compare the minimal number of sensors selected by the different methods to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set is used. We use the training dataset of Dataset II, III-A and III-B (described in Section 2.5) as the preliminary dataset to perform sensor selection using the different sensor selection methods to select sensor subsets for the corresponding subject. More specifically, for our SLES method, this preliminary dataset is used to train a $OSLN_{(S)}$ and calculate $score_j$ in each iteration of our SLES method (see Algorithm 1). For the sensor selection methods C_1 , C_2 , C_3 , C_4 , C_5 , C_6 , C_7 , C_8 , C_9 , and C_{10} (see Table 5.3), this preliminary dataset is used to calculate the P300 detection accuracy (for C_1 and C_6), the average character spelling accuracy across all epochs (for C_2 and C_7), AUC (for C_3 and C_8), the C_{cs} score (for C_4 and C_9), and the SSNR score (for C_5 and C_{10}). For the sensor selection methods C_{11} and C_{12} (see Table 5.3), this preliminary dataset is used to train CCNN (for C_{11}) and BN3 (for C_{12}) in order to analyze the weights of the trained CCNN and BN3 to select appropriate sensor subsets from the initial sensor set. After using different sensor selection methods to select sensor subsets, we calculate the spelling accuracy of the aforementioned OCLNN-based P300 speller, EoCNN-based P300 speller, and SVM-based P300 speller with the selected sensor subsets. The training datasets of Dataset II, III-A and III-B are used to train the classifier used in the aforementioned P300 speller implementations with the selected sensor subsets. Then, the test dataset of Dataset II, III-A and III-B are used to calculate the spelling accuracy of the aforementioned P300 speller implementations with the selected sensor subsets. The spelling accuracy is calculated using Equation (5.3). In this equation, $acc^m_{char(k)}$ denotes the spelling accuracy when using the first k epochs for each character and using the EEG signals acquired with the selected sensor subset containing m number of sensors. $N^m_{tc(k)}$ denotes the number of truly predicted characters when using the first k epochs for each character and using the EEG signals acquired with the selected sensor subset containing m number of sensors, and S_c denotes the number of all characters in the evaluation dataset. After the evaluation of the spelling accuracy, the minimal number of sensors needed to acquire EEG signals for epoch k is calculated as m_{min}, where m_{min} is the minimal $m \in [1, 63]$ which makes $acc^m_{char(k)} >= acc^{64}_{char(k)}$.

$$acc^{m}_{char(k)} = \frac{N^{m}_{tc(k)}}{S_c} \tag{5.3}$$

The setup for our SLES algorithm (Algorithm 1) is the following. The input to SLES is $S = \{s_1, s_2, ..., s_j, ... s_C\}$ and E_s. We set C=64 because the datasets used in the experiments are recorded with 64 sensors. We set E_s=4. For detailed discussion why E_s=4 see Section 5.4.1. SLES uses $OSLN_{(S)}$ as the ranking function. $OSLN_{(S)}$ uses the input tensor ($N \times |S|$). N = 240 because the signal sampling frequency is 240 Hz and we take each individual pattern to be the signal samples between 0 and 1000 ms posterior to the beginning of each intensification.

5.3.2 Experimental Results

Table 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11 and 5.12 show the minimal number of sensors selected by the different sensor selection methods to acquire EEG signals while keeping the spelling accuracy the same as the accuracy achieved when the initial large sensor set of all 64 sensors is used. The first column in the tables lists the different selection methods we compare. Each row provides the minimal number of sensors selected by a method to acquire EEG signals for different epoch numbers $k \in [1, 15]$. A number in bold indicates that the minimal number of sensors selected by the corresponding method is the lowest among all methods. Overall, the minimal number of sensors selected by our SLES method is lower than the minimal number of sensors selected by all other methods in most cases. SLES is able to reduce the minimal number of sensors selected by other methods with up to 44 sensors.

For the P300 speller with our CNN-based classifiers, i.e., OCLNN and EoCNN, (see Table 5.4, 5.5, 5.6, 5.7, 5.8, and 5.9), in 83 out of 90 cases, the minimal number of sensors selected by our SLES is lower than the minimal number of sensors selected by all other methods. Our SLES is able to reduce the minimal number of sensors selected by other methods with up to 44 sensors. The largest reduction occurs when comparing the minimal number of sensors selected by SLES with the minimal number of sensors selected by C_8 on epoch number $k = 7$ for Dataset III-A using the OCLNN-based P300 speller.

For the P300 speller with the SVM-based classifier (see Table 5.10, 5.11 and 5.12), in 41 out of 45 cases, the minimal number of sensors selected by our SLES is lower than the minimal number of sensors selected by all other methods. Our SLES is able to reduce the minimal number of sensors selected by other methods with up to 40 sensors. The largest reduction occurs when comparing the minimal number of sensors selected by SLES with the minimal number of sensors selected by C_{12} on epoch number $k = 2$ for Dataset III-B.

Finally, our SLES method is robust because: 1) SLES is effective in reducing the number of sensors when the P300 speller is implemented with different classifiers. From Table 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11 and 5.12, we can see that no matter the P300 speller is implemented with CNN-based classifier or SVM-based classifier, the minimal number of sensors selected by SLES is lower than the minimal number of sensors selected by all other methods in most cases; 2) SLES is effective when it is used for different subjects, i.e., no matter that SLES is used with Dataset III-A, Dataset III-B or Dataset II, the minimal number of sensors selected by SLES is lower than the minimal number of sensors selected by all other methods in most cases.

Table 5.4: Minimal number of sensors selected by different methods for Dataset II. The P300 speller is implemented using the CNN-based classifier OCLNN.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**11**	**11**	**46**	**15**	**22**	11	**9**	**3**	**3**	**3**	**3**	4	**3**	**3**	**3**
C_1	32	30	47	39	38	22	16	6	6	6	6	6	6	7	6
C_2	17	20	55	19	34	15	12	6	5	3	4	**3**	6	7	7
C_3	18	18	47	18	30	10	12	6	5	**3**	5	6	6	6	6
C_4	24	27	49	17	28	**8**	10	7	6	7	4	**3**	**3**	5	6
C_5	40	33	48	41	38	35	20	9	8	9	9	8	9	9	8
C_6	48	33	49	47	32	30	28	28	20	15	10	20	10	15	15
C_7	44	32	49	48	31	27	27	27	22	18	10	10	8	10	10
C_8	44	36	50	38	33	32	25	25	10	10	10	10	10	17	12
C_9	45	35	44	34	34	34	25	25	10	17	18	17	15	15	15
C_{10}	48	35	49	44	40	30	33	29	21	19	20	20	20	18	17
C_{11}	25	25	54	24	24	14	15	15	10	10	12	17	15	15	15
C_{12}	29	27	59	25	31	22	15	18	13	13	15	18	19	21	18

5.4 Discussions

In this section, we discuss the configuration of input parameter E_s in SLES (see Algorithm 1). Also, we discuss the impact of different CNN architectures on selecting sensors.

Table 5.5: Minimal number of sensors selected by different methods for Dataset III-A. The P300 speller is implemented using the CNN-based classifier OCLNN.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**17**	**50**	**50**	**25**	**40**	**35**	**20**	**50**	**60**	**20**	**50**	**35**	**21**	**16**	35
C_1	27	64	58	59	55	64	43	58	64	29	57	49	29	29	53
C_2	29	64	64	60	59	59	36	60	**60**	31	56	56	31	28	60
C_3	28	64	64	60	64	60	55	64	64	36	64	55	39	39	46
C_4	53	64	63	64	64	64	60	64	61	61	64	56	55	37	61
C_5	30	64	56	64	55	64	43	63	64	31	64	57	36	33	55
C_6	30	63	64	64	64	64	57	64	64	61	64	53	53	53	59
C_7	44	57	64	64	64	64	62	64	64	55	64	53	59	52	51
C_8	49	59	60	63	64	64	64	63	64	60	64	56	56	54	56
C_9	22	64	56	55	58	64	56	56	64	34	52	48	27	36	**34**
C_{10}	34	63	64	64	64	64	59	64	64	61	64	58	61	53	59
C_{11}	25	64	56	55	58	64	56	56	64	34	54	48	27	36	39
C_{12}	28	64	64	56	64	64	58	56	64	41	64	53	39	36	41

5.4.1 Configuration of E_s in SLES

In this section, we show how we configure the input parameter E_s in SLES. We use only the preliminary dataset of Dataset III-A and use the OCLNN-based P300 speller implementation to show the experiments on how to tune E_s because we obtain the same E_s value when we perform experiments using all datasets and using all the aforementioned P300 speller implementations to tune E_s for SLES. We divide the preliminary dataset of Dataset III-A into two parts. The first part contains 60% of the preliminary dataset of Dataset III-A. The second part contains the left 40% of the preliminary dataset of Dataset III-A. The first part, i.e., the 60% of the preliminary dataset of Dataset III-A, is used to train $OSLN_{(S)}$ (see Section 5.2.2) while running SLES with different E_s configurations, i.e., E_s=1, 2, 4, 8, 16, 32 and 64. With each E_s configuration, SLES selects a set of sensor subsets for Dataset III-A. The second part, i.e., the left 40 % of the preliminary dataset of Dataset III-A, is used to evaluate the spelling accuracy of the aforementioned P300 speller implementation with the selected sensor subsets. The P300 spelling accuracy is calculated using Equation (5.3). Then, we

Table 5.6: Minimal number of sensors selected by different methods for Dataset III-B. The P300 speller is implemented using the CNN-based classifier OCLNN.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**25**	**17**	**31**	**27**	**26**	**21**	**27**	**32**	**37**	**32**	**31**	**29**	**34**	**29**	**23**
C_1	32	23	**31**	28	31	24	32	40	43	44	38	32	42	35	26
C_2	28	24	52	38	36	25	35	45	41	41	**31**	30	42	38	31
C_3	26	29	42	50	35	23	32	54	49	48	38	**29**	39	38	33
C_4	31	25	61	40	45	24	38	57	60	57	59	55	49	47	43
C_5	44	33	36	34	31	24	34	41	43	48	38	42	42	37	26
C_6	50	56	55	49	48	58	48	50	50	50	48	44	51	49	45
C_7	52	48	49	49	50	49	49	59	48	48	48	46	45	47	46
C_8	49	49	54	52	49	54	48	59	49	49	49	44	46	44	44
C_9	44	56	49	49	40	25	35	39	39	49	51	37	**34**	35	42
C_{10}	49	52	55	44	50	51	46	50	52	51	49	50	51	49	50
C_{11}	44	56	49	49	40	25	35	49	39	49	51	37	38	34	42
C_{12}	54	61	59	55	52	27	30	42	45	63	54	48	47	34	42

calculate the minimal number of sensors m_{min} for the different E_s configurations as described in Section 5.3.1. In this experiment, we divide the preliminary dataset of Dataset III-A into two parts, i.e., one part containing 60% of the preliminary dataset of Dataset III-A and one part containing the left 40% of this dataset because the majority of the researchers use this ratio to split a dataset [XMYR16, VELB18, LLJ$^+$18].

The experimental results are shown in Table 5.13. The first column in the table lists the different configurations of E_s in SLES. Each row provides the minimal number of sensors selected by SLES for different epoch numbers $k \in [1, 15]$. A number in bold indicates that the minimal number of sensors selected by SLES with the corresponding E_s configuration is the lowest compared with the minimal number of sensors selected by SLES with other E_s configurations. From Table 5.13, we can see that, in most cases, the minimal number of sensors selected by SLES with E_s=4 is the lowest compared to the minimal number of sensors selected by SLES with other E_s configurations. Therefore, we set E_s=4 when using SLES.

Table 5.7: Minimal number of sensors selected by different methods for Dataset II. The P300 speller is implemented using the CNN-based classifier EoCNN.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**10**	**16**	**28**	**19**	**22**	9	**8**	**3**	**3**	**3**	**3**	4	**3**	**3**	**3**
C_1	28	33	40	41	38	19	15	6	6	6	6	6	6	7	6
C_2	15	24	48	**19**	34	14	12	6	5	**3**	4	**3**	6	7	7
C_3	14	20	41	20	30	9	10	6	5	**3**	5	6	6	6	6
C_4	19	29	43	21	28	**7**	9	7	6	8	4	**3**	**3**	5	6
C_5	31	34	42	43	38	31	17	9	8	9	9	8	9	9	8
C_6	43	35	44	49	33	29	25	27	20	15	10	20	10	15	15
C_7	39	32	45	52	31	27	26	27	22	18	10	10	8	11	10
C_8	37	36	47	40	33	30	25	25	10	10	10	10	10	17	12
C_9	42	37	43	36	34	33	24	25	10	18	18	17	15	15	15
C_{10}	43	38	45	47	40	28	32	29	21	19	20	20	20	18	15
C_{11}	19	29	48	25	23	13	12	15	10	10	11	17	15	15	15
C_{12}	26	31	53	28	31	18	13	18	13	13	15	18	19	19	18

5.4.2 Exploring the Impact of the CNN Architecture on Sensor Selection

We perform experiments to explore the impact of different CNN architectures on the sensor selection process in order to address the issue mentioned in the third paragraph of Section 5.1 and the issue mentioned at the end of Section 5.2.3. The P300 speller implemention used for this experiment is the CNN-based classifier OCLNN. We use the preliminary dataset of Dataset III-A to train our OSLN, OTLN (proposed and presented in Section 4.1.2 of Chapter 4), OCLNN (proposed and presented in Chapter 3), CCNN [CG11], and BN3 [LWG$^+$18]. We select sensor subsets by directly analyzing the weights of the convolution layer of OSLN, OCLNN, and OTLN, as well as select sensor subsets by directly analyzing the weights of the first convolution layer of CCNN and BN3 (as done in [CG11] and [LWG$^+$18]). We use the evaluation dataset of Dataset III-A to evaluate the P300 spelling accuracy of the aforementioned P300 speller implementation with the selected sensor subsets. Then, we calculate the minimal number of sensors m_{min} selected by analysing the weights of OSLN, OCLNN, OTLN, CCNN, and BN3. For the detailed calculation of m_{min} see Section 5.3.1.

Table 5.8: Minimal number of sensors selected by different methods for Dataset III-A. The P300 speller is implemented using the CNN-based classifier EoCNN.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**18**	**39**	**37**	**21**	**41**	**38**	**20**	**39**	**44**	**23**	**46**	**33**	**21**	**16**	**31**
C_1	27	51	44	55	55	60	43	46	50	30	52	46	29	29	50
C_2	30	53	50	56	59	55	36	49	52	35	51	54	31	27	56
C_3	30	53	50	56	64	56	55	52	53	39	59	53	39	39	41
C_4	55	52	48	59	63	60	60	52	49	62	60	52	55	37	56
C_5	31	54	51	59	55	60	43	50	50	36	59	55	36	33	50
C_6	31	52	50	58	64	59	57	53	52	62	61	50	54	53	56
C_7	44	56	52	55	62	60	61	51	53	58	60	50	59	52	48
C_8	50	57	47	52	64	60	64	50	52	61	60	51	56	54	53
C_9	23	55	43	51	55	58	56	45	51	40	48	42	27	36	**31**
C_{10}	34	56	49	53	64	60	59	64	52	62	60	53	61	53	52
C_{11}	26	53	42	50	58	61	56	46	52	39	51	44	26	36	36
C_{12}	30	60	48	55	64	63	58	52	53	44	59	48	39	36	40

The experimental results are shown in Table 5.14. The first column in the table lists the different CNNs. Each row provides the minimal number of sensors selected by analysing the weights of the different CNNs for different epoch numbers $k \in [1, 15]$. A number in bold indicates that the minimal number of sensors selected by analysing the weights of the corresponding CNN is the lowest, compared to the minimal number of sensors selected by analysing the weights of other CNNs.

Table 5.14 shows that the minimal number of sensors selected by analysing the weights of our proposed one-convolution-layer CNNs, i.e., OSLN, OCLNN, as well as OTLN, is lower than the minimal number of sensors selected by analysing the weights of CCNN and BN3. The reason is the following. CCNN and BN3 have multiple convolution layers. The information needed for proper sensor selection is distributed over the weights of all convolution layers. In CCNN and BN3, only the weights of the first convolution layer are used for analysis and proper sensor selection because the weights of the other convolution layers can hardly be used for proper sensor selection. This is because the information for the importance of each sensor in the sensor set is stored in the input neurons of the input tensor to a CNN. This

Table 5.9: Minimal number of sensors selected by different methods for Dataset III-B. The P300 speller is implemented using the CNN-based classifier EoCNN.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**25**	**26**	**28**	**27**	**23**	**14**	**16**	**25**	**30**	**29**	33	**22**	30	**29**	**23**
C_1	31	30	**28**	28	27	18	23	34	37	40	37	29	39	35	26
C_2	28	32	48	38	32	20	24	38	35	39	**31**	27	39	38	31
C_3	26	40	38	50	33	21	22	47	42	45	36	26	37	38	33
C_4	31	31	57	40	42	20	25	50	53	52	56	40	40	47	43
C_5	44	39	33	34	29	22	26	34	36	46	38	36	38	37	26
C_6	50	60	50	49	45	42	37	45	45	49	46	38	47	49	45
C_7	52	55	48	49	46	39	38	52	40	47	47	41	41	45	46
C_8	49	55	51	52	48	45	37	53	42	46	45	37	41	44	44
C_9	44	62	46	49	38	23	24	35	33	45	50	32	**29**	36	42
C_{10}	50	60	52	44	49	41	35	44	46	48	49	41	47	49	50
C_{11}	44	60	48	49	37	23	25	36	35	42	49	30	35	34	41
C_{12}	54	64	56	55	45	26	28	42	41	51	52	38	40	34	42

input tensor is directly related to the first convolution layer of CCNN and BN3 by directly connecting each receptive field of the input neurons in the input tensor with each neuron in the first layer of CCNN and BN3. These connections are expressed by their corresponding weight in the first convolution layer. Thus, the weights of the first convolution layer of CCNN and BN3 can be used for analysis and sensor selection. However, the weights of the other convolution layers of CCNN and BN3 only express the connections of the neurons of the first convolution layer with the neurons of the other convolution layers. We can hardly build any direct relation between the input tensor (that stores the information for the importance of each sensor in the sensor set) with the neurons in the other convolution layers of CCNN and BN3 by using the weights of these layers. As a result, the weights of the other layers of CCNN and BN3 can hardly be used for analysis and proper sensor selection. Therefore, CCNN and BN3 cannot use all the information available for proper sensor selection. In contrast, our OSLN, OCLNN, OTLN have only one convolution layer. All the information needed for sensor selection is captured by the weights of the single convolution layer of OSLN, OCLNN, and OTLN. We analyse the weights of the single convolution layer

Table 5.10: Minimal number of sensors selected by different methods for Dataset II, The P300 speller is implemented using the SVM-based classifier ESVM [RG08].

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**13**	**10**	**36**	**20**	**31**	**16**	**8**	**4**	**3**	**5**	**3**	**5**	**3**	**6**	**3**
C_1	30	32	42	38	43	28	15	8	6	9	7	10	6	8	6
C_2	25	28	40	29	40	19	16	9	6	11	**3**	11	7	**6**	8
C_3	21	23	38	30	34	25	13	8	7	10	6	14	7	7	7
C_4	26	24	**36**	33	38	18	14	10	8	12	5	13	4	9	9
C_5	43	29	37	40	41	36	24	11	9	13	8	12	10	16	11
C_6	50	38	52	50	50	33	26	30	18	19	11	24	12	14	16
C_7	47	40	55	48	49	29	29	29	16	21	13	23	11	17	13
C_8	50	37	54	42	46	31	30	29	16	23	13	20	16	16	19
C_9	51	35	49	39	54	39	29	31	19	25	14	19	13	19	17
C_{10}	49	44	50	40	50	30	31	26	20	21	19	19	19	20	16
C_{11}	28	30	56	30	36	19	17	16	9	20	16	19	16	17	12
C_{12}	31	33	61	36	38	23	18	20	11	24	20	21	21	26	14

in OSLN, OCLNN, and OTLN to select sensors. Thus, OSLN, OCLNN, and OTLN use all the information available for proper sensor selection compared to CCNN and BN3. As a result, OSLN, OCLNN, and OTLN can select more appropriate sensor subsets and further reduce the minimal number of sensors needed to acquire EEG signals without losing spelling accuracy. The aforementioned discussion explains the issue mentioned in the third paragraph of Section 5.1.

The experimental results in Table 5.14 also explain why in Section 5.2.3, our ranking function to rank the sensors in the sensor set is based on OSLN instead of OCLNN and OTLN. When compared with OTLN, the minimal number of sensors selected by analyzing the weights of OSLN is lower than the minimal number of sensors selected by analyzing the weights of OTLN. Therefore, our ranking function in SLES is based on OSLN instead of OTLN. When compared with OCLNN, the minimal number of sensors selected by analyzing the weights of OSLN is comparable with the minimal number of sensors selected by analyzing the weights of OCLNN. However, the network complexity of OSLN (8,722 parameters) is only 51.67% of the complexity of OCLNN (16882 parameters). This means that the time of training the OSLN is much

Table 5.11: Minimal number of sensors selected by different methods for Dataset III-A, The P300 speller is implemented using the SVM-based classifier ESVM [RG08].

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**19**	**44**	**54**	**38**	**40**	**42**	33	56	**58**	**39**	**44**	**27**	**35**	**21**	**36**
C_1	31	56	58	49	45	57	44	**54**	64	43	55	51	46	32	56
C_2	32	53	57	48	44	59	47	55	64	43	50	53	47	32	55
C_3	29	53	**54**	48	47	56	35	55	63	45	52	57	42	36	49
C_4	56	50	53	44	42	55	**31**	**54**	60	56	49	51	43	37	57
C_5	31	55	58	50	45	49	43	60	64	46	58	48	47	33	57
C_6	35	60	61	57	54	56	52	64	64	53	62	59	57	50	62
C_7	43	59	61	55	49	57	52	64	64	50	62	58	55	50	62
C_8	47	54	64	57	48	54	55	62	64	59	57	52	56	49	63
C_9	33	55	60	51	48	62	59	63	64	56	53	54	39	46	49
C_{10}	33	62	62	59	53	57	57	64	64	54	61	60	59	52	55
C_{11}	29	61	59	54	55	59	56	58	64	46	53	45	41	40	37
C_{12}	35	64	63	56	59	61	60	59	64	51	58	47	44	46	43

smaller than the time of training the OCLNN. Thus, the speed of SLES with the ranking function based on OSLN is much higher than the speed of SLES with the ranking function based on OCLNN (see Line 2 to Line 4 in Algorithm 1). Therefore, our ranking function in SLES is based on OSLN instead of OCLNN.

5.5 Conclusions

In this chapter, we propose a novel sensor selection method, called SLES, for reducing the number of sensors needed to acquire EEG signals for a P300 speller without losing spelling accuracy. SLES uses an iterative parameterized backward elimination algorithm to eliminate and select sensors and it uses our novel $OSLN_{(S)}$ as a ranking function to evaluate the importance of a sensor. Our SLES is also robust across different P300 speller implementations and different subjects. Experimental results show that the minimal number of sensors selected by our SLES method is lower than the minimal number of sensors selected by other methods in most cases. Therefore, our SLES can further reduce the cost and power consumption of the P300 speller, thereby

Table 5.12: Minimal number of sensors selected by different methods for Dataset III-B, The P300 speller is implemented using the SVM-based classifier ESVM [RG08].

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SLES	**18**	**19**	**34**	**29**	**17**	31	**39**	**41**	**22**	**36**	**24**	33	**33**	**26**	**27**
C_1	28	27	43	39	32	35	45	50	36	51	36	**30**	45	37	29
C_2	27	25	42	38	29	31	**39**	54	36	49	40	**30**	44	30	**27**
C_3	24	31	45	35	26	30	42	49	34	44	38	32	47	**26**	**27**
C_4	23	26	59	36	27	**29**	44	52	35	42	37	36	39	37	30
C_5	35	29	44	40	29	33	48	50	39	49	38	33	43	40	31
C_6	49	58	56	51	44	40	57	61	48	59	46	42	57	51	47
C_7	41	52	53	48	46	48	53	63	50	54	49	44	52	47	44
C_8	40	50	53	49	47	50	49	60	47	50	41	44	51	50	46
C_9	42	51	48	47	41	37	52	61	43	49	45	39	46	44	45
C_{10}	42	54	51	52	38	43	57	61	44	55	50	46	53	50	49
C_{11}	43	55	49	47	34	33	45	53	40	52	51	39	49	29	46
C_{12}	52	59	56	61	46	41	51	64	49	64	56	41	60	30	46

facilitating the utilization of P300 spellers into people's daily life.

Table 5.13: Minimal number of sensors selected by SLES with different E_s configurations.

	Epochs														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E_s=1	27	56	58	57	50	60	44	53	63	30	55	33	24	19	46
E_s=2	18	**45**	52	34	43	46	28	**48**	58	26	47	30	20	16	**36**
E_s=4	**15**	46	**49**	**28**	**40**	**36**	**18**	**48**	59	**21**	**46**	**27**	**19**	**13**	38
E_s=8	19	54	**49**	31	41	48	26	**48**	**57**	25	**46**	34	20	**13**	37
E_s=16	22	54	55	42	45	56	39	53	63	29	53	35	25	20	37
E_s=32	25	59	57	47	49	58	48	54	63	32	54	37	28	23	41
E_s=64	27	60	60	48	50	60	55	55	64	36	59	42	31	24	44

Table 5.14: Minimal number of sensors selected by analysing different CNNs.

	Epochs														
CNN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
OSLN	23	**60**	**55**	**49**	51	**60**	**53**	55	64	**31**	53	**46**	**27**	**32**	**39**
OCLNN	**22**	**60**	**55**	**49**	**49**	**60**	**53**	**53**	**63**	32	**52**	**46**	**27**	33	**39**
OTLN	23	62	56	51	53	62	**53**	56	64	34	53	**46**	**27**	33	**39**
CCNN	25	64	56	55	58	64	56	56	64	34	54	48	**27**	36	**39**
BN3	28	64	64	56	64	64	58	56	64	41	64	53	39	36	41