MIE498 Thesis Final Report

Dahong Zhang

1004743307

Supervisor: Xilin Liu

Table of Content

Project 1: OpenBCI Classification for the Closed and Opened Eyes

Abstract

Alpha wave is a well known brain activity for human beings during an eyes-closed resting condition, and Brain Computer Interface (BCI) allows the alpha brain waves to be converted into computer commands. In this report, we present a study on visualizing and detecting alpha brain waves by self-collecting Electroencephalography (EEG) signals with OpenBCI devices. The study aims to evaluate the effectiveness of using the alpha wave feature to classify between open-eye and close-eye signals. To achieve this objective, the data was preprocessed with a 0~30 Hz Bandpass filter and Fast Fourier Transform (FFT) calculation. After that, we visualized the difference between the two classes of signal in frequency domain and applied KNN, Random Forest Classifier (RFC), and Neural Network models. The result shows that the alpha wave feature only presents in the close-eye signal, and the RFC achieved the highest accuracy of 82%. It suggests that the EEG brain signal under open-eye and close-eye conditions can be classified with the alpha wave feature, and the RFC is the best model for classifying the self-collected EEG data.

1.0 Background

A Brain Computer Interface (BCI) enables communication between the human brain and external devices by converting brain signals into computer commands. With this technology, individuals can express their thoughts using their brain activity, without any physical movements. The most widely researched type of BCI is Electroencephalography (EEG) based, which uses noninvasive techniques to acquire and measure brain signals. EEG-based BCIs are used in this experiment due to their ease of recording with low-cost equipment. Many studies have shown that the BCI is able to capture the alpha activity, which is the neural oscillations in the frequency range 8~12 Hz and is one of the most frequently observed EEG activities in individuals [1]. The alpha wave is prominent during a resting state with eyes closed and is suppressed in the presence of visual stimulation.

This project aims to visualize and detect the alpha activity by self-collecting EEG data and implementing signal processing techniques. Additionally, various machine learning models are also applied to the preprocessed data to compare the classification accuracy. For data processing, a study in 2007 investigating the EEG differences between open and closed eyes has suggested that a bandpass of 0–30 Hz is effective for filtering out the noise [1]. In terms of the EEG classification, another study in 2019 has compared the performance of classifying the EEG data by using K-Nearest Neighbor (KNN), Decision Tree (DT), and Neural Network [2]. The result from this study has shown that Neural Network has the highest accuracy of classification, followed by Decision Tree, and then KNN. However, as our self-collected EEG dataset is different from the study of 2019 and the model complexity of KNN and Decision Tree outperform Neural Network, our project will apply all of these models and compare their performance with our datasets.

2.0 Data Collection

The EEG signal is collected by OpenBCI devices, including handset and Cyton+Daisy biosensing board with 8 channels, and OpenBCI GUI. The data is collected with two separated sets: open eyes and close eyes. Each set is collected for 1 minute, and the sampling frequency is 250Hz. In this experiment, we only collected the data for one participant. For the open eye set, the participant is required to keep their eyes open for 1 min with minimal blinks once the OpenBCI starts to collect EEG signals. Similarly, for a closed eye set, the participant is asked to close their eyes for 1 min. Open eye set is labeled as 0 while the close eye set is labeled as 1. All 8 channels are used when collecting the EEG data.

3.0 Method

Data Preprocessing

By plotting the signal for all 8 channels, we observed that a great amount of noises are picked up because there are no obvious patterns in the graph. Therefore, we need to apply filters to clean up the noise. By applying the bandpass filter with 0 and 30 Hz cutoff frequency to all channels, the curves for both datasets become flat, and the data fluctuate around 0 volts. Furthermore, Gert et al. 2012 has mentioned that the alpha wave activity mainly occurs in the prefrontal area of the brain [3], so we only choose the data collected from the prefrontal channel (Fp1, Fp2) for the following data preprocessing.

By observation on the zoom-in graph for each channel, we found that the data in channel 1 best matches with the open and close eye pattern of EEG (see Figure 1). To prepare the data for the classifier training and testing, each of the open and close eye dataset are divided into 12 pieces of 5 sec data, so there will be 24 data pieces in total. In particular, the 24 pieces are shuffled and further divided into 6 folds for cross validation, which means there will be 20 training data and 4 testing data in each iteration. By applying Welch's method [4], the Power Spectral Density (PSD) for each data piece in the frequency domain is calculated.

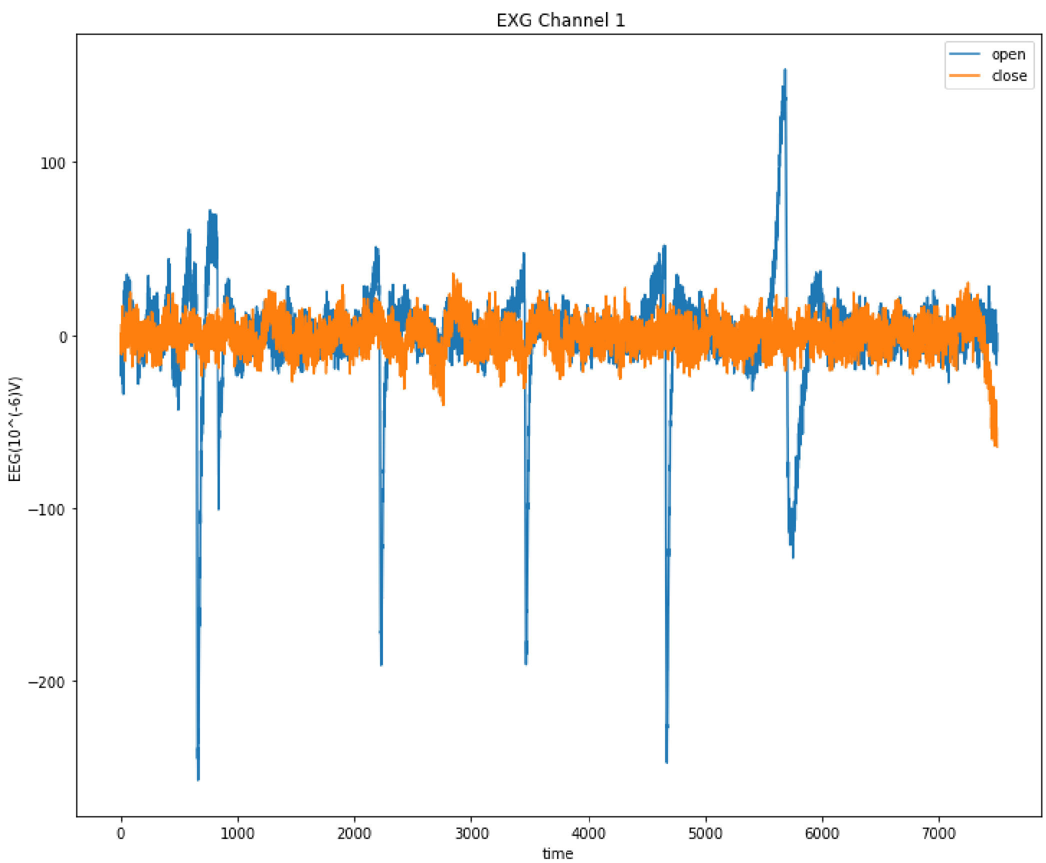

Figure 1: Channel 1 capturing open and close eye signals

Classifiers Setup

KNN model, Random Forest Classifier (RFC) model and Neural network are developed and compared. The number of neighbors in KNN is set up as 3. The parameters in the RFC model are set up as n_estimators=50. The neural network is set up with two fully connected layers: the first layer with 12 neurons and the second layer with 8 neurons. 'Relu' is the activation function between each layer. The model is trained by minimizing mean squared error with SGD optimizer.

4.0 Result

By using the log-log plot for Power Spectrum Density (PSD) versus frequency, we can clearly observe the alpha wave at around 10 Hz for close-eye data in Figure 2, while there are not any obvious peaks for open-eye data.

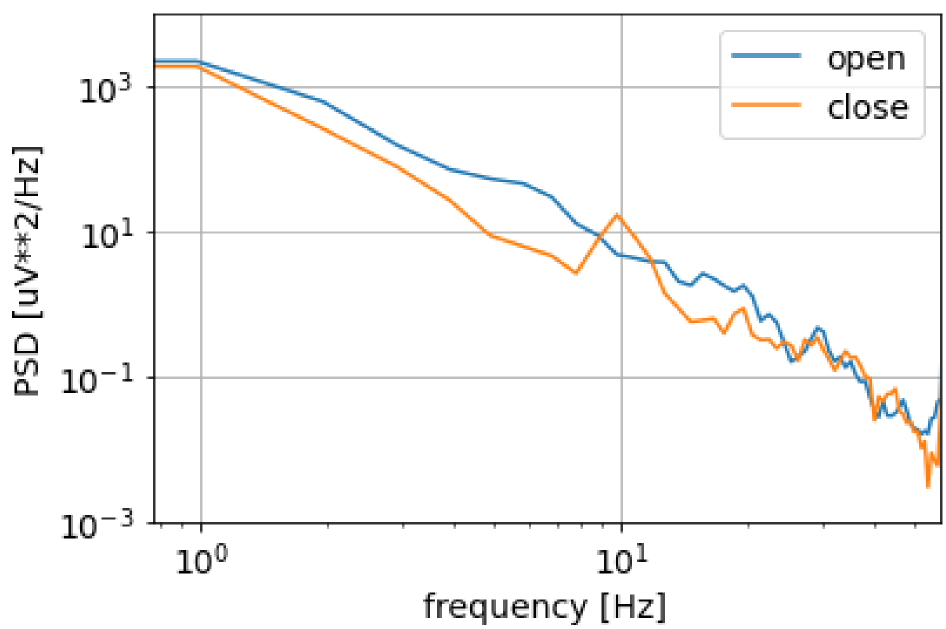

Figure 2: open and close eye Power Spectrum Density (PSD) in frequency domain

The accuracy of classifying the signal above for each model is summarized in Table 1.

Model	Accuracy
KNN	79.2%
Random Forest	82.0%
Neural Network	78.3%

Table 1: model accuracy of classifying open and close eye

5.0 Conclusion

The visualization obtained shows that the alpha wave feature only presents in close-eye brain state but is absent in open-eye brain state. The summary of model performance also indicates that the random forest model is the best model for classifying the collected EEG signals. However, this is the result contributed by only one participant. To make our conclusion generalized, the inclusion of EEG data from more participants is needed.

Project 2: Channel Reduction for P300 Speller Experiment

Abstract

This report explores Electroencephalography (EEG) based Brain-Computer Interfaces (BCIs) for P300 character spelling tasks. It discovers how the P300 response, the largest event-related potential (ERP) observed in EEG recordings, is detected with a binary classification task. Deep learning models, particularly Convolutional Neural Networks (CNNs), have been found to be effective in classifying the P300 response in EEG signals. This study reuses the state-of-the-art model, One Convolution Layer Neural Network (OCLNN), to compare the impacts of different channel selection methods. By filtering 64 channels down to the most effective 8 channels, 4 channels, and further to a single channel, the P300 classification accuracy reduces from 92% to 89%, 88%, and 84% respectively. In terms of character spelling accuracy, reducing the number of channels to 8 did not increase the effort required to attain 100% accuracy, while reducing it to 4 channels needed two additional epochs per character to get 100%. However, reducing to only one channel fails to achieve 100% character spelling accuracy within 15 episodes.

1.0 Introduction

A Brain Computer Interface (BCI) enables communication between the human brain and external devices by converting brain signals into computer commands. With this technology, individuals can express their thoughts using only their brain activity, without any physical movements. The most widely researched type of BCI is Electroencephalography (EEG) based, which uses noninvasive techniques to acquire and measure brain signals. EEG-based BCIs are popular due to their ease of recording with low-cost equipment.

BCI has been developed to help locked-in (e.g. Amyotrophic Lateral Sclerosis (ALS)) patients [5]. In recent years, BCI has also been popularly developed for healthy people, in application domains such as entertainments [6], mental state monitoring [7] as well as in IoT services [8]. In the past decades, character spelling has been one of the most popular applications. Current character spellings are based on these P300, SSVEP and motor imagery (event-related (de) synchronization (ERD/ERS)). In this report, we will focus on the character spelling task base on the P300 response.

In recent years, numerous deep learning models, Convolution Neural Network (CNN) particularly, have been proven to be effective for classifying the P300 response in EEG signal. We will summarize the architecture of different state-of-the-art models in the following literature review sections. Most of these models are built and evaluated based on 64 channels EEG data. Using 64 channels for classification has been shown to improve speller accuracy; however, with the consideration of increased system cost and setup time, and also the ease of further implementation on the hardware, the number of channels in use should be optimized. The study published by Krusienski et al. (2008) has proposed the most effective 8 channels for enhancing P300 speller experiment [9]. In addition, Colwell et al. (2014) further verified the effectiveness of these 8 channels by using different channel selection methods and testing with different subjects [10].

This report applies one of the existing CNN models called OCLNN with 64 channels EEG data as a benchmark, and then reduces the input to 8, 4 and 1 channels to see how the classification performance and speller accuracy changes.

2.0 Literature Review

2.1 P300 Detection

The largest event-related potential (ERP), known as the P300 signal, was initially discovered by Sutton and colleagues in 1967 [11]. The P300 signal is observed in EEG recordings as a positive deflection in voltage at a latency of approximately 300 ms after a rare stimulus, as illustrated in Figure 3. The detection of the P300 signal is considered as a binary classification task, where one class corresponds to the presence of the P300 signal within a specific time window, while the other class corresponds to its absence within the same period.

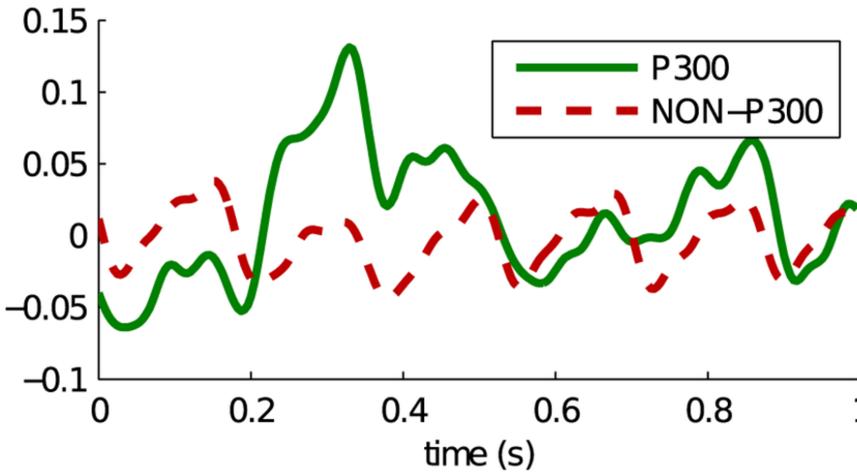

Figure 3: Plots of P300 and non-P300 EEG signal

2.2 Deep Learning Model for P300 classification

OCLNN

One Convolution Layer Neural Network (OCLNN). Shan et al. proposed a one-convolutional-layer architecture [12], which is the simplest CNN-based architecture for P300 detection in terms of number of layers. It consists of a 1D convolutional layer with a ReLU activation function, followed by a Softmax classification layer. The convolutional layer divides the temporal signals from the input channels into 15 parts and performs a convolution operation for temporal and spatial feature extraction. Dropout is applied before the Softmax classification layer. The structure of the model is shown in Figure 4.

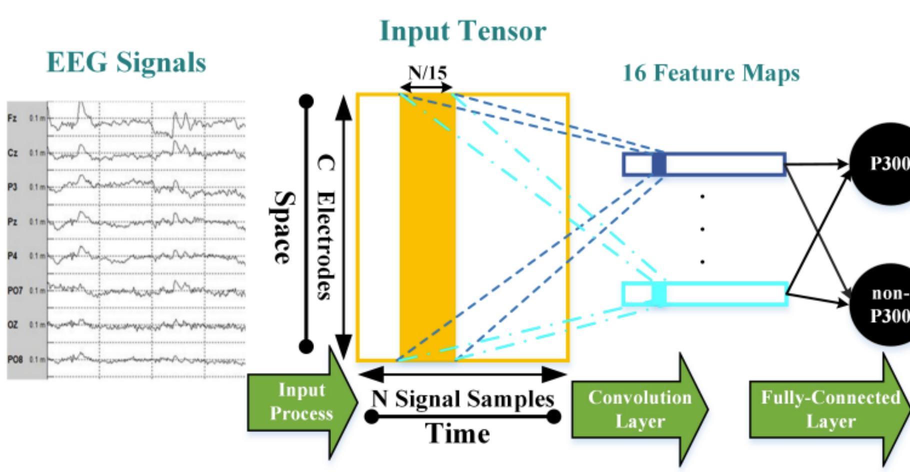

Figure 4: Structure of OCLNN

SepConv1D

Montserrat et al. (2021) studies the performance of existing CNN architectures with diverse complexities for single-trial within-subject and cross-subject P300 detection on four different datasets [13]. The study proposed SepConv1D, a very simple CNN architecture consisting of a single depthwise separable 1D convolutional layer followed by a fully connected Sigmoid classification neuron. It was found that with as few as four filters in its convolutional layer and an overall small number of parameters, SepConv1D obtained competitive performances in the four different datasets. The structure of the model is shown in Figure 5.

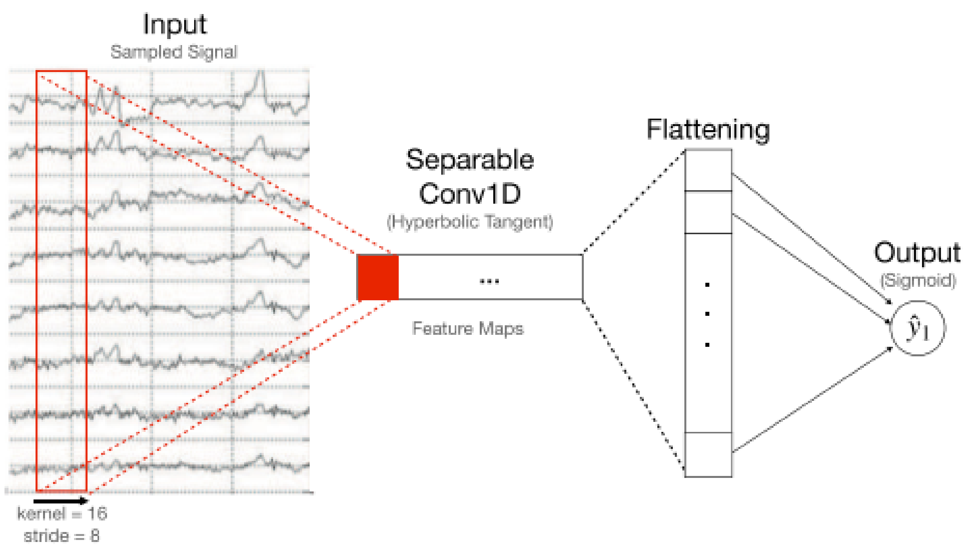

Figure 5: Structure of SepConv1D

2.3 State-of-the-art EEG Channel Selection Methods

Standard Channel

Krusienski et al. (2008) compares the effects of spatial channel selection, channel referencing, data decimation, and maximum number of model features and employs stepwise linear discriminant analysis (SWLDA) to build a classifier [9]. By incorporating posterior locations in addition to classical P300 recording locations, the study shows that SWLDA, along with favorable parameters derived from offline comparative analysis, can significantly enhance the online classification performance of P300 speller responses. Using a SWLDA classifier, both offline and online results obtained from 64-channel data show that some of the most discriminable EEG features evoked by the P300 speller occur at posterior electrodes (PO7, PO8, Oz), and that these features can significantly improve classification performance when used in conjunction with the classical P300 feature space (Fz, Cz, Pz). By unioning these two sets of electrodes, the result proposed the channel set {Fz, Cz, Pz, Oz, PO7, PO8} to be the most effective for improving P300 spelling classification.

Active Channel Selection

In 2014, another study by Colwell et al. employed generalized standard feature-selection methods and introduced a new channel selection method called jump-wise regression [10]. They study each method using real P300 Speller data, and show that active channel selection can enhance speller accuracy for most users compared to a standard channel set, especially for those who experience low performance using the standard set. Among the methods tested, jumpwise regression provides similar accuracy gains as the best-performing feature-selection methods and is robust enough for online use. The results also show that the optimal electrodes chosen by this new method have a great overlap with the standard channel set proposed by Krusienski et al. (2008).

3.0 Datasets Preprocessing

This study uses the BCI Competition II - Dataset IIb for validation and testing purposes. The BCI competition was held by Benjamin Blankertz in 2003 and the purpose of the competition is to explore different signal processing and classification methods for BCI [14]. The dataset represents a complete record of P300 evoked potentials recorded with BCI2000 using a 6 by 6 character matrix (See Figure 6) described by Donchin et al., 2000, and originally by Farwell and Donchin, 1988.

In this experiment, the participant is asked to focus on each character in a word sequence (i.e. one character at a time) that was prescribed by the investigator. Each row and column is then intensified randomly and successively at a rate of 5.7Hz. In particular, two of the twelve intensifications contain the target character (i.e. one in a target row and the other in a target column), Since the row/column containing the targeted character is a rare stimulus, the P300 signal is evoked in the participant's brain. By detecting the P300 signal, we can infer which row or column the participant is focused on, and then locate the target character with the predicted row and column positions.

Figure 6: P300 Character Spelling Matrix

The BCI2000 platform was used to record the EEG signal in this dataset. The brain signals were collected from 64 electrodes at a sampling frequency of 240Hz. Each intensification lasted for 100ms, followed by a 75ms blank period for the matrix. The experiment consisted of 15 epochs for each character and there are 12 intensifications for each epoch (i.e. there are 12 * 15 = 180 intensification for each character). The index number of each intensification is depicted in Figure 7. After each sequence of 15 epochs, the matrix was blank for 2.5s to signal the subject that the character was completed and to shift focus to the next character. Figure 8 demonstrates the flow of the experiment.

Figure 7: P300 Character Spelling Matrix with Row/Column Index

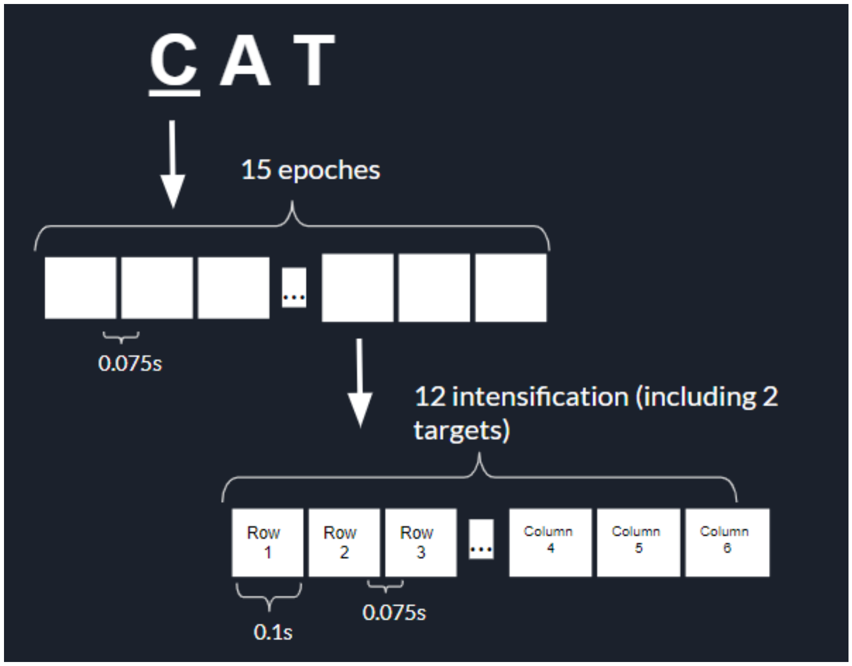

Figure 8: Experiment Setup and Data Format in BCI Competition II

In this dataset, there is only one subject divided into separate training and test datasets. The size of data in each set is summarized in Table 2. Each sample of data is a 2 by 2 matrix with the shape of (N x C), where N is the number of temporal samples in 1s after each intensification, C is the number of electrodes used for EEG signal collection (i.e. in this study, N =240, C = 64) Each individual data is bandpass filtered between 0.1Hz and 20 Hz to remove high frequency noise and normalized to have zero mean and unit variance.

Dataset	**Characters**	**P300**	**non-P300**
training	42	42 x 15 x 2 = 1260	42 x 15 x 10 = 6300
testing	31	31 x 15 x 2 = 930	31 x 15 x 10 = 4650

Table 2: model accuracy of classifying open and close eye

4.0 Methodology

4.1 Network Architecture

In this study, we reuse the OCLNN model structure proposed by Shan et. al (2018) [12]. With data from all 64 channels as the predictor, the convolution layer divides the (240,64) input signals into 15 segments, performing convolution operations on each segment to acquire features. Figure 9 demonstrates how the convolution works. The kernel size of the convolution operation is (240/15, 64), and each receptive field in the input tensor consists of a (240/15, 64) tensor of signal samples. In the time domain, these signal samples are derived from a time period of 1/15 s, while in the spatial domain, they originate from all 64 electrodes. By converting each receptive field of data into a feature map, the convolution operation in this layer enables the learning of features from both raw temporal and raw spatial information. No overlapped convolution is used, so the stride for the convolution operation is 240/15 = 16. After the convolution, Rectified Linear Unit (ReLU) is applied as an activation function to model a neuron's output in this layer, as networks with ReLUs are trained more quickly than those with traditional activation functions. Dropout is employed in this layer to reduce overfitting, with the dropout rate set to 0.25. This layer produces 16 feature maps.

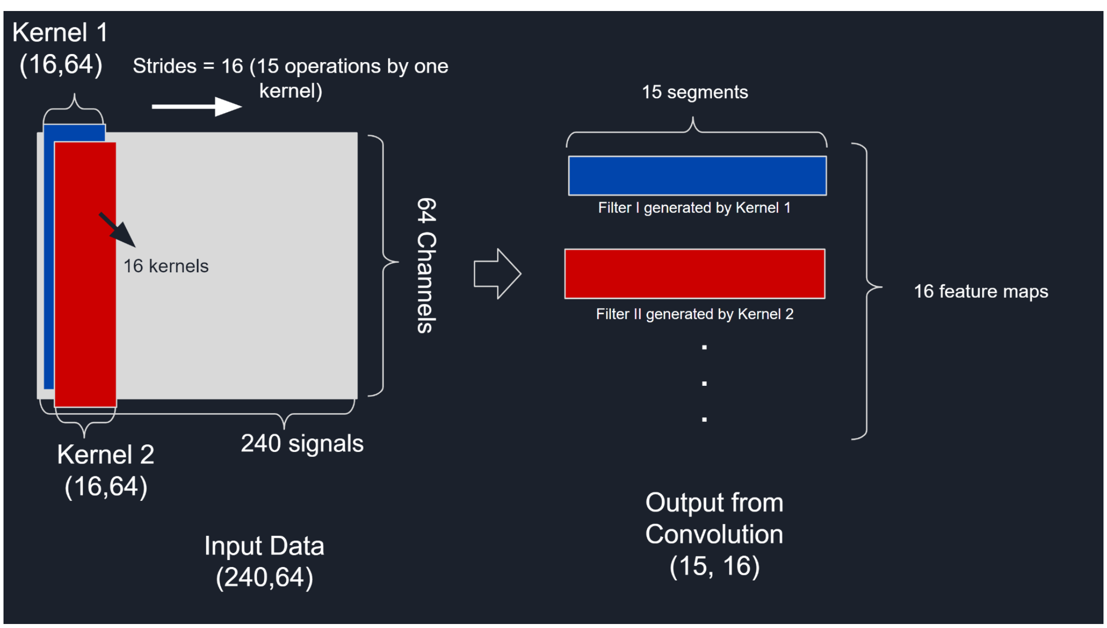

Figure 9: Detailed Demonstration of the OCLNN Convolution Layer

The output layer of OCLNN employs a fully-connected operation with two neurons: one representing the "P300" class and the other representing the "non-P300" class. Figure 10 demonstrates how the fully-connected layer works. This operation correlates the 16 feature maps from Layer 1 with these two classes. Softmax is utilized as the activation function for the neurons in this layer, with the output of the Softmax function for the "P300" and "non-P300" classes denoted by P1(i,j) and P0(i,j), respectively. P1(i,j) indicates the probability of detecting a P300 signal, while P0(i,j) indicates the probability of not detecting a P300 signal at epoch i and intensification j. In Figure 10, Equation 1 defines the detection of a P300 signal, where X(i,j) represents the input tensor being classified and E is the binary classifier.

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

Figure 10: Equation for classifying P300 signal

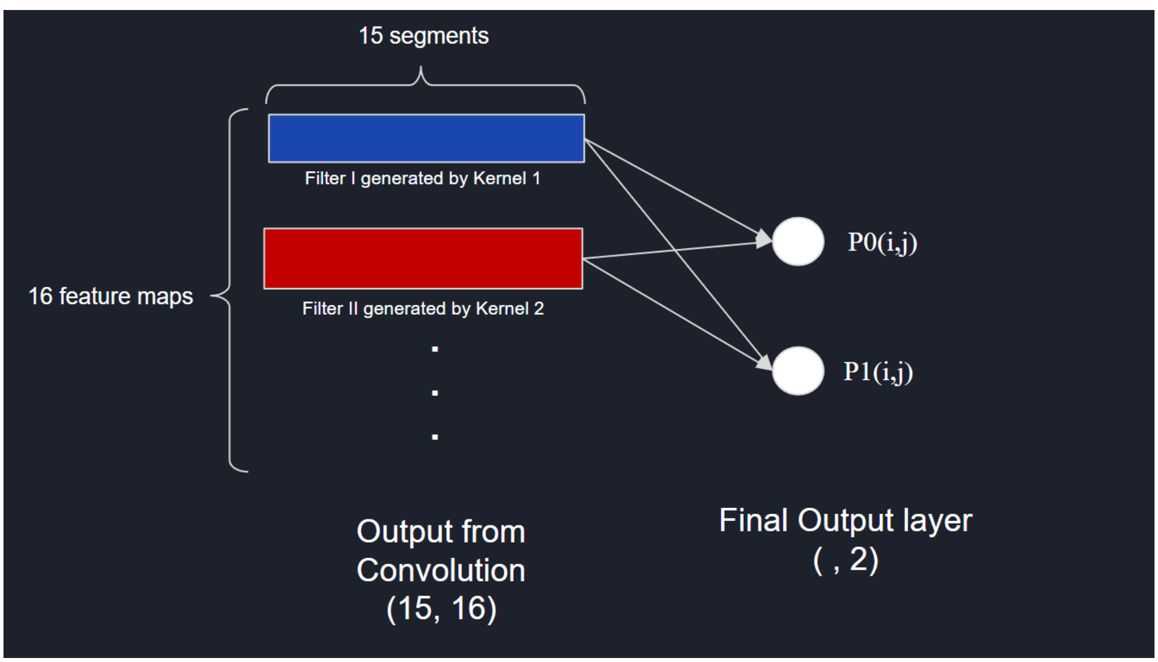

Figure 11: Detailed Demonstration of the OCLNN Fully-Connected Layer

The architectures of OCLNN with 64, 8, 4, and 1 are illustrated from Figure 12.1 to Figure 12.4, respectively. The first column describes the sequence of the layers. The second column describes the shape of the output from each layer. The last column describes the number of parameters for each layer, and the sum is at the bottom of the table. For OCLNN with 64 channels input, we use the formula ((width of the filter * height of the filter + 1)*number of filters) to calculate the number of parameters for the convolution layer. In this formula, the "+1" calculates the number of biases for each filter. The shape of the filter is 16 by 64, and the number of filters generated is 16. Therefore, our number of parameters in this layer is (16 x 64 + 1) x 16 = 16400. For the fully connected layer, we use the formula ((width of the filter * height of the filter + 1)*number of neurons). Again, the "+1" in the formula is the number of biases for each neuron. The shape of the filters is 15 by 16, and the number of neurons is 2. Therefore, our number of parameters in this layer is (15 x 16 + 1) x 2 = 482. In particular, dropout and flatten layers do not generate extra learnable parameters so their number of parameters are 0. Therefore, the total parameters for OCLNN with 64 channels are 16400 + 482 = 16882. The OCLNN models with other channels input use similar methods to calculate the parameters. For the model with 8, 4 and 1 channels, the total number of parameters are 2546, 1522, and 754, respectively.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 15, 16)            16400

 dropout (Dropout)           (None, 15, 16)            0

 flatten (Flatten)           (None, 240)               0

 dense (Dense)               (None, 2)                 482

 flatten_1 (Flatten)         (None, 2)                 0

=================================================================
Total params: 16,882
Trainable params: 16,882
Non-trainable params: 0
_________________________________________________________________
```

Figure 12.1: Number of Model Parameters in Each Layer for 64 Channels

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 15, 16)            2064

 dropout (Dropout)           (None, 15, 16)            0

 flatten (Flatten)           (None, 240)               0

 dense (Dense)               (None, 2)                 482

 flatten_1 (Flatten)         (None, 2)                 0

=================================================================
Total params: 2,546
Trainable params: 2,546
Non-trainable params: 0
_________________________________________________________________
```

Figure 12.2: Number of Model Parameters in Each Layer for 8 Channels

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 15, 16)            1040

 dropout (Dropout)           (None, 15, 16)            0

 flatten (Flatten)           (None, 240)               0

 dense (Dense)               (None, 2)                 482

 flatten_1 (Flatten)         (None, 2)                 0

=================================================================
Total params: 1,522
Trainable params: 1,522
Non-trainable params: 0
_________________________________________________________________
```

Figure 12.3: Number of Model Parameters in Each Layer for 4 Channels

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 15, 16)	272
dropout (Dropout)	(None, 15, 16)	0
flatten (Flatten)	(None, 240)	0
dense (Dense)	(None, 2)	482
flatten_1 (Flatten)	(None, 2)	0

Total params: 754
Trainable params: 754
Non-trainable params: 0

Figure 12.4: Number of Model Parameters in Each Layer for 1 Channel

4.2 Character Spelling

To reproduce the targeted character, we use the method proposed by Shan et al 2000 [12]. We use the P1(i,j) output of the model above for the "P300" class to determine the position of the target character in the P300 speller matrix. To accomplish this, we employ Equation 2, 3, and 4, where *C(j)* represents the total probabilities, *indexcol* and *indexrow* indicate the column and row index of the target character in the matrix presented in Figure 7, respectively. j refers to a column intensification when $j \in [1, 6]$, while j represents a row intensification when $j \in [7, 12]$. Equation 2 computes the cumulative probability of a P300 signal being evoked by the intensification j across all epochs (n = 15). In Equation 3, we assign the index of the maximum C(j) to *indexcol* when $j \in [1, 6]$. This equation identifies the index of the column intensification with the highest sum of probabilities that generated a P300 signal, which corresponds to the column position of the target character. Equation 4 calculates the row position of the target character in the same way as Equation 3. The target character's position in the matrix shown in Figure 13 is determined by the intersection of its row and column positions.

$$C_{(j)} = \sum_{i=1}^{n} P^{1}_{(i,j)} \tag{2}$$

$$index_{col} = \underset{1 \leq j \leq 6}{argmax}\, C_{(j)} \tag{3}$$

$$index_{row} = \underset{7 \leq j \leq 12}{argmax}\, C_{(j)} \tag{4}$$

Figure 13: Equations to Calculate Targeted Row and Column

4.3 Channels Selection

In this project, we tried to reduce the total 64 EEG channels to 8, 4 and 1 channels as input to the model and discussed how they impacted the accuracy. For the selection of 8 channels, we directly used the standard set proposed by Krusienski et al. (2008) [9], which is {Fz, Cz, P3, Pz, P4, PO7, PO8, Oz} indicated with red circles in Figure 14. For the selection of 4 channels, Colwell et al. mentioned that in most of the cases, the channels set {Pz, PO7, PO8, Oz}contains the most discriminable information for classification, but the results may vary across different subjects and dataset. Therefore, in the experiment, we try the positions {Pz, PO7, PO8, Oz} indicated with cyan circles in Figure 11. To further reduce the input to 1 channel, Pz, PO7 and Oz are usd for the prediction separately. Due to the similarity of symmetric channels (e.g. PO7 and PO8), we do not use PO8 for the following implementation.

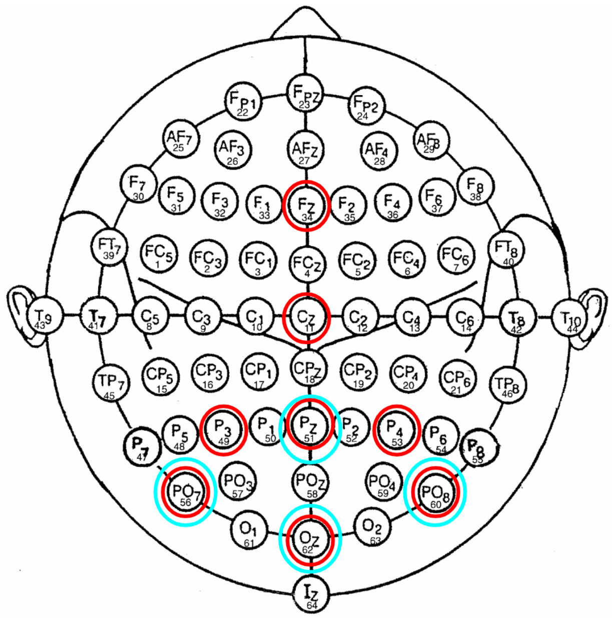

Figure 14: Subset of EEG Channels Selected in this Project - the Selected 8 Channels Circled in Ced; the Selected 4 Channels Circled in Cyan

5.0 Result

This project will use two metrics to measure the success of the channel reduction for the P300 speller experiment: the P300 classification accuracy and character spelling accuracy. The P300 classification accuracy refers to the sum of the true positive and true negative divided by the total number of data in the testing dataset. True negative means the number of truly classified P300s for a test dataset, while true negative denotes the number of truly classified non-P300s for the test dataset. Character spelling accuracy refers to the number of truly predicted characters in the test dataset divided by the total number of characters in the testing dataset. For most of the models, 100% character spelling accuracy can be achieved by using less than equal to 15 epochs of data for each character. As a result, in order to compare the performance of different models, we determine the minimum number of epochs of data required per character for a model to attain 100% character spelling accuracy.

5.1 P300 Classification Accuracy

Table 3 shows the P300 Classification Accuracy for the two state-of-the-art models with simple structure and strong classification performance. The OCLNN model proposed by Shan et. al

(2018) and SepConv1D by Montserrat et al. (2021) both use BCI Competition II - Dataset IIb for testing. OCLNN achieves 92% of P300 classification accuracy, while SepConv1D gets 91%.

Model	OCLNN	SepConv1D
P300 Classification Accuracy	92%	91%

Table 3: P300 Classification Accuracy for different state-of-the-art models

Table 4 summarizes the P300 classification accuracy of the OCLNN model in classifying the BCI Competition II - Dataset IIb testing data using 64, 8, 4, and single EEG channels as inputs. The model achieves the highest accuracy of 92% when all 64 channels are utilized. However, reducing the number of channels to 8 results in a 3% drop in accuracy, while reducing it to 4 channels results in a 4% decrease in accuracy. Moreover, using only the PO7 channel as predictor decreases the accuracy to 84%, while using the Pz channel and the Oz channel further reduces the accuracy to 82%.

Number of Channels Used		P300 Classification Accuracy for OCLNN
64 channels		92%
8 channels (Fz, Cz, P3, Pz, P4, PO7, PO8, Oz)		89%
4 channels (Pz, PO7, PO8, Oz)		88%
1 channel	**PO7**	84%

	Pz	82%
	Oz	82%

Table 4: P300 Classification Accuracy when Using 64, 8 and 4 Channels for OCLNN Model

5.2 Character Spelling Accuracy

Table 5 summarizes the character spelling accuracy of the OCLNN model using 64, 8, and 4 EEG channels as inputs. The utilization of 64 and 8 channels requires 9 epochs of data per character to achieve 100% accuracy. Furthermore, reducing the number of channels to 4 needs 11 epochs of data per character to converge. However, reducing the number of channels to 1 fails to get 100% within 15 epochs. For each single channel, the PO7 channel can have the highest character spelling accuracy of 55%, followed by 29% for the Oz channel, and then the 26% for the Pz channel.

Number of Channels Used	**Character Spelling Accuracy (%)/Epochs**														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
64 channels	83	87	96	96	96	96	96	96	100	100	100	100	100	100	100
8 channels	70	77	80	96	96	93	93	93	100	100	100	100	100	100	100
4 channels	70	74	80	96	96	90	93	93	96	96	100	100	100	100	100
1 channel PO7	25	35	38	52	48	52	48	55	52	42	52	55	48	55	55
1 channel Pz	6	6	10	13	23	16	16	19	29	25	23	29	23	29	26

1 channel Oz	10	10	16	19	12	19	23	16	19	23	19	16	26	26	19

Table 5: Character Spelling Accuracy per Epoch when Using 64, 8 and 4 Channels for OCLNN Model

6.0 Discussion and Conclusion

According to the findings presented in Table 4, reducing the number of channels for classification using the OCLNN model leads to a reduction in classification accuracy, albeit within a range of less than 5%. Moreover, the results in Table 5 suggest that decreasing the number of channels as a predictor for character spelling tasks increases the number of epochs to obtain character spelling accuracy.

In this experiment, the result is subject specific (e.g. for one subject and one dataset). To generalize the conclusion, it is necessary to include cross-subject datasets in the next stage of research. As outlined in Section 2.2, many other studies have tested their models using BCI Competition III - Dataset II subjects A and B, as well as BNCI Horizon 2020. It is recommended to begin with BCI Competition III - Dataset II A and B, as it has the same data format and experimental setup. The only difference is the testing characters. However, the BNCI Horizon 2020 dataset may require additional effort to process the data due to its different experimental settings and data format. Furthermore, while this project only used the OCLNN model, it would be worthwhile to explore other deep learning models, such as SepConv1D with few filters in the convolutional layer, and evaluate the impact on accuracy. Last but not least, it is important to explore additional subsets of channels for detecting the P300 signal and reproducing spelling characters. While the application of the four channels subset proposed by Colwell et al. (2014) seems effective in our results, there may be other subsets of channels that could produce better results. Therefore, the next step should be to identify the optimal set of EEG channels in order to maximize model performance.

Reference

[1] Y. Li and J. Li, "A framework for automatic generation of test items from concept maps," Computers & Education, vol. 50, no. 4, pp. 1339-1350, May 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1388245707004002. [Accessed: Apr. 06, 2023].

[2] S. Kumar and V. Singh, "A Survey on Deep Learning Techniques for Named Entity Recognition," in Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 143-148, doi: 10.1145/3325773.3325785.

[3] R. D. Luecht, L. J. Hitchcock, et al., "Integrating a cognitive model into an adaptive educational hypermedia system," Computers & Education, vol. 58, no. 1, pp. 246-259, Jan. 2012. [Online]. Available: https://doi.org/10.1016/j.compedu.2011.08.019.

[4] R. Craig, "Real-time obstacle avoidance for manipulators and mobile robots," in Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, Apr. 2005, pp. 118-125, doi: 10.1109/ROBOT.1986.1087539.

[5] S. Downes, "Exploring the use of blogs as learning spaces in the higher education sector," Australasian Journal of Educational Technology, vol. 21, no. 2, pp. 232-247, May 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1388245705004608.

[6] A. Khemmar et al., "A Brain–Computer Interface to a Plan-Based Activity Scheduler," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 26, no. 6, pp. 1143-1153, Jun. 2018. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/55058113/A_Brain-Computer_Interface_to_a_Plan-Bas20171120-2453-1lvt8lk-libre.pdf.

[7] A. M. Boripuntakul et al., "Efficacy of cognitive interventions in elderly: A systematic review," International Journal of Geriatric Psychiatry, vol. 29, no. 6, pp. 551-560, Jun. 2014. doi: 10.1002/gps.3790.

[8] A. J. Koester et al., "Assessing the Comfort of Wearable Computers," in Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, Oct. 2014, pp. 1917-1922, doi: 10.1109/SMC.2014.6974208.

[9] K. A. Mutlu, A. K. Çakır, C. Güzeliş and İ. Öztekin, "An FPGA-based real-time motor unit number estimation system," in Journal of Neuroscience Methods, vol. 167, no. 2, pp. 308-316, 2008, doi: 10.1016/j.jneumeth.2007.08.016.

[10] S. Suresh, S. Sundaram and K. N. Venugopal, "A review of EEG based brain–computer interface for assistive technologies," in Journal of Medical Systems, vol. 38, no. 4, pp. 23, 2014, doi: 10.1007/s10916-014-0023-0.

[11] [Sutton et al., 1967] Samuel Sutton, Patricia Tueting, et al. Information delivery and the sensory evoked potential. Science, 155(3768):1436–1439, 1967.

[12] Shan, H., Liu, Y., & Stefanov, T. P. (2018, July). A Simple Convolutional Neural Network for Accurate P300 Detection and Character Spelling in Brain Computer Interface. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13-19 July 2018, 1604-1610.

[13] Alvarado-Gonzalez, A. M., Fuentes-Pineda, G., & Cervantes-Ojeda, J. (2021). A few filters are enough: convolutional neural network for P300 detection. Neurocomputing, 425, 37-52.

[14] [Blankertz, 2003] B Blankertz. BCI competition II webpage. http://www.bbci.de/competition/ii/, 2003.