Biosignal Processing and Modeling SS2024

PRACTICAL PROJECT
Artifact detection: Maze game with Emotiv device

Chutong Ren, Wenlong Li from Group Mind Reader

July 17, 2024

Keywords Brain–computer interface · EEG · artifacts detection · maze game control

1 Introduction

Brain-computer interface (BCI) technology enables direct communication between the brain and external devices. Electroencephalography (EEG) is a crucial component for BCIs, as it is the most commonly utilized method for recording brain electrical activities non-invasively along the scalp. However, EEG data is notorious for its low signal-to-noise ratio (SNR), as it is easily contaminated by noise and artifacts. These artifacts can arise from various sources, such as eye movements (electrooculography, EOG), muscle activity (electromyography, EMG), and heartbeats (electrocardiography, ECG). The typical amplitudes for signals of interest are usually around 10 μV, whereas EEG artifacts often have much higher magnitudes, around 100 μV. Traditionally artifacts are viewed as noise that must be reduced to enhance the clarity of EEG data. However, there were several studies in which artefacts especially EOG was used to control certain device[4] or games[5]. In our research, we adopt a novel approach by directly using artifacts as features to control movement in a maze game. This method not only simplifies the processing pipeline but also explores new possibilities for the use of artifacts in BCI applications.

2 Materials and Methods

The hardware schematic/system structure is shown in **Fig. 1** (Left), which includes Emotiv Epoc X headset (hardware) for data collection, Emotiv Xavier ControlPanel (software) for impedance detection and OpenViBE (software) for data recording. Meanwhile, the classification results of artifacts are used to control the maze game on the screen.

The flow chart is depicted in **Fig. 2**, which consists of five modules: data acquisition, preprocessing, feature extraction, artifact detection and maze game control. Our system is in real-time, meaning multiple tasks can occur in parallel, such as data acquisition, data processing, filtered-data plotting, and maze game control. To handle this, we used multiprocessing in Python to run these functions concurrently and multiprocessing.Queue to transfer data between different processes. As long as the queue is not empty, items are retrieved and processed.

2.1 Data Acquisition

All data were recorded using a Emotiv EPOC X (EMOTIV, San Francisco, U.S.A.) headset with 14 Channel (AF3, F7, F3, FC7, T7, P7, P8, T8, FC6, F4, F8, AF4. And CMS/P3, DRL/P4 are

Author: Chutong Ren, Wenlong Li
Enrollment number: 03782591, 03729544
E-mail: chutong.ren@tum.de, wenlong.li@tum.de

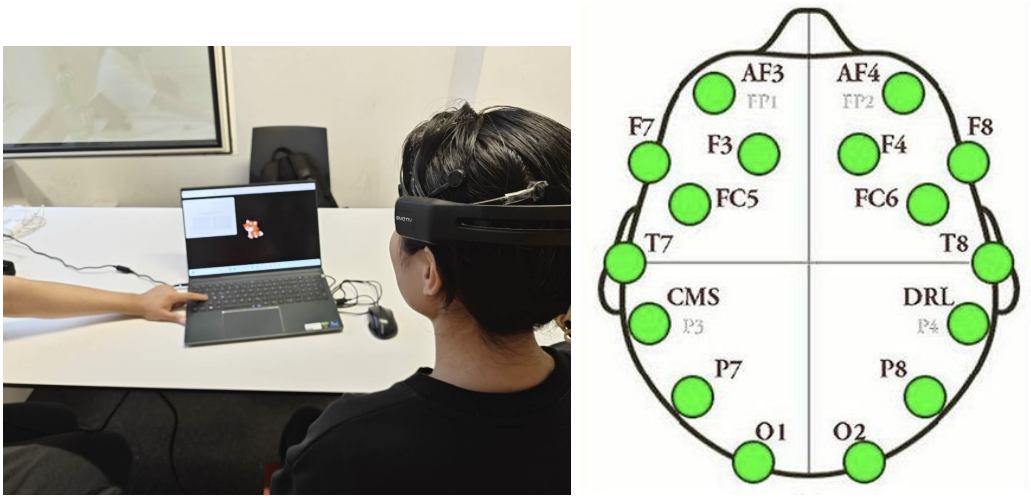

Fig. 1: Experimental setup. Left: Experimental environment. The subject wears a Emotiv headset and sits in front of the computer screen with a Maze Game. Right: EEG electrode distribution over the scalp following the 10–20 System.

for reference.) placed according to a 10-20 system, which can be seen in **Fig. 1**(right). Note that: AF4 electrode is non-working/a bad channel, so it's excluded and not considered in this experiment, although it plays a crucial role in classifying right eye blinking.

After charging the headset, hydrate the sensors/electrodes as thoroughly as possible to ensure good contact quality. Then connect the USB Bluetooth and wear the headset. Observe the Emotiv Xavier ControlPanel to adjust the fit for optimal placement.**??** When all indicators turn green, the impedance between the sensors and the scalp is low enough, then we can then start recording data. The sampling frequency of the headset was 128 Hz. The headset possessed an in-built Notch filter at 50 and 60 Hz with 16 Bit resolution.

To acquire real-time data, OpenViBE2.0 acquisition server was used in this experiment, as shown in **Fig. 3** (Left). The EEG data were synchronized using 'pylsl.StreamInlet', and all functionalities of our system were implemented using Python in VSCode. We set an overlap of 0.75 second between each 1-second time window to ensure continuous data analysis and smooth transitions between windows.

2.2 Data Preprocessing

The EEG data preprocessing pipeline usually includes filtering, detrending, epoching, baseline correction, ICA, re-referencing, and downsampling. However, we performed only filtering for real-time artifact classification, which requires less processing time and higher efficiency. After acquiring live data, we applied filtering to remove noise. Specifically, data was first low-pass filtered using a Chebyshev Type II filter with cut-off frequency as 29 Hz. This was followed by high-pass filtering with the same filter type, using cut-off frequency as 1 Hz. The cut-off frequency was chose to best adapt to the needs.

2.3 Feature Extraction

Channels T7 and T8 are crucial for detecting chewing, typically associated with the beta frequency band (13-30 Hz)[2]. In contrast, AF3 and AF4 are vital for recognizing blinking, which occurs predominantly within the delta band (1-3 Hz)[6][3]. However, in our case, the AF4 channel was broken and as a subtitute, F4 channel was introduced to extract feature of RightEyeBlink. Therefore, the signal power was computed using the Fast Fourier Transform (FFT) for these four channels within their respective frequency bands to extract features for each time window of data.

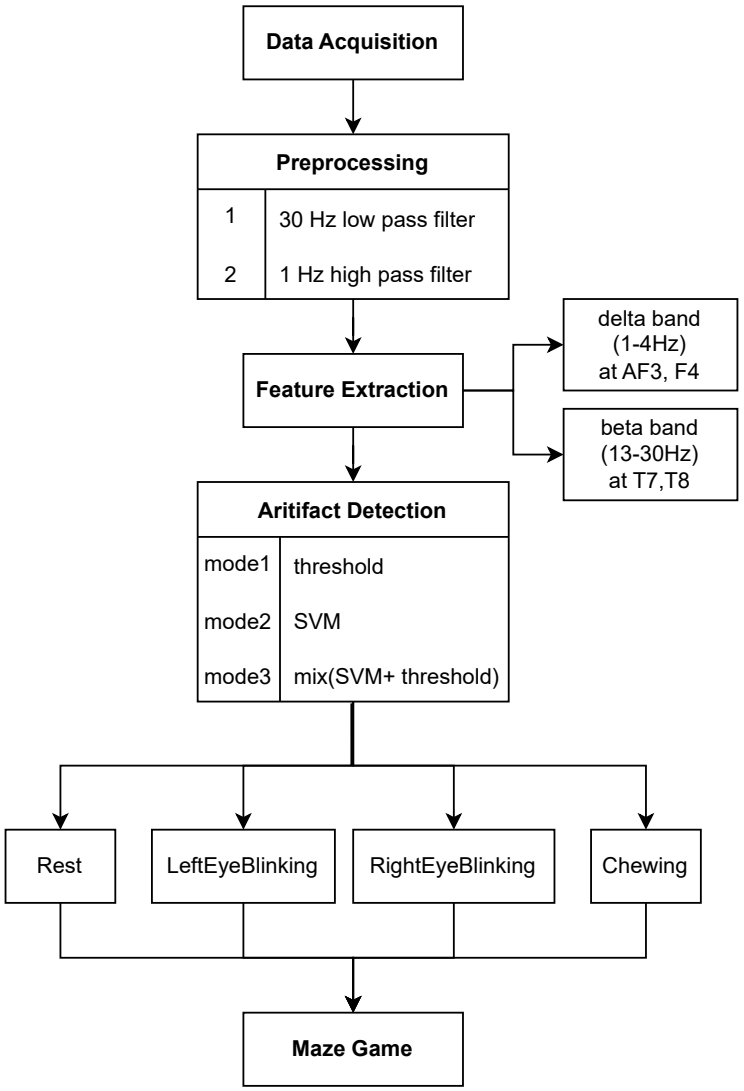

Fig. 2: Flow chart of real-time artifact detection-induced maze game system.

2.4 Classification models & Artifacts detection

Before acquiring a model to classify three artifacts and the rest state, we needed to collect training datasets to fit classification model. The experimental design for training data acquisition is described in **Fig. 4**. We went through two sessions in different days. Each session was divided into four blocks. Each block conducted one of the following four different tasks: left eye blinking, right eye blinking, chewing and stay rest. In between, a 5-minute subject-controlled break was implemented. Each block contains 40 trials, with each trial lasting 2 seconds. During each trial, when an instruction (a fox) appeared on the screen, the subject performed the corresponding action. The training scenario was displayed in **Fig. 2.2 Left**

After collecting eight training datasets, we preprocessed the data as described in **Sec. 2.2**. We then used these datasets, which contained 160×5 features and 160×1 labels, to fit an SVM model. Finally, we saved the trained model using joblib in Python.

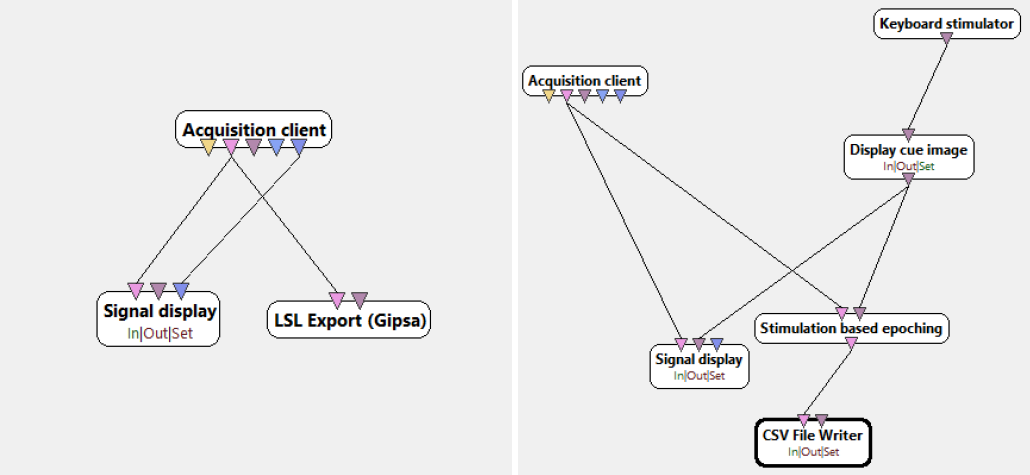

Fig. 3: Left: OpenViBE designer configuration pipeline under real-time recording scenario. Right: OpenViBE designer configuration pipeline under offline recording scenario.

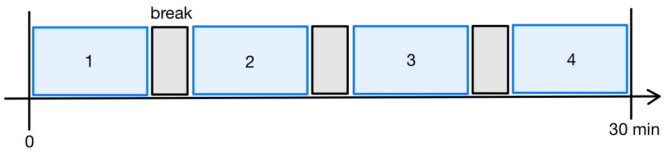

Fig. 4: Offline experimental design. The experiment was divided into four blocks. In between, a subject-controlled break was implemented. The total duration of an experiment is around 30 min.

2.5 Maze game developement

The maze game interface was designed using the PyGame package, as depicted in **Fig. 5**. In the game, the goal was to control the subject (yellow square) to the destination (red square). The control of the subject was accomplished by following matches:

- Right: RightEyeBlink
- Left: LeftEyeBlink
- Up: Chewing
- Still: Rest

When three consecutive predicted actions are consistent, then the movement is executed in the maze game. This ensures that the game's response is based on consistent and accurate artifact classification.

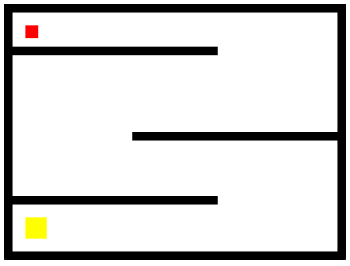

Fig. 5: Maze Game. The task of pilot was to move the subject (yellow square) to the destination (red square).

3 Results

To evaluate the effectiveness and quality of the offline collected datasets, the Event-Related Potentials (ERP) of four events (RightEyeBlink, LeftEyeBlink, Chewing and Rest) were computed. The feature extraction only involved four relevant channels (F4, AF3, T7 and T8), and the corresponding ERPs of these four channels were depicted in **Fig. 6**. It is worth noticing that the datasets were pre-selected to achieve the best performance. Furthermore, due to the fact that not all channels were considered at each event, irrelevant channels were represented by dashed lines for each event. Solid lines revealed the actual relevant channels. The x-axis describes the time after the onset of events (0 s).

The explanations of the four different events were shown following:

- **Chewing:** The T8 and T7 channels exhibited large fluctuations ($\approx 20\mu V$) with relatively high frequency after the onset of chewing cues. The similar waveforms of T8 and T7 corresponded to the symmetry of these two channels. It could also be observed that the activities of the F4 and AF3 channels increased as well but with lower amplitude.
- **LeftEyeBlink:** It could be concluded that the blinking of the left eye induced great response in channel AF3 ($\approx 15\mu V$), leading to a large low frequency waveform. Additionally, the other three channels were less influenced by the event than AF3.
- **Rest:** The waveforms of 4 channels were roughly similar during rest. However, the peak value of AF3 was unexpectedly large ($\approx 6\mu V$). This revealed the fact that the recording conditions might vary with experiment sessions, leading to unexpected errors.
- **RightEyeBlink:** The waveform of F4 was slightly larger than that of AF3. The 'irrelevant' T7 and T8 released stronger responses. Moreover, there was no apparent difference between the amplitudes of RightEyeBlink ($\approx 4\mu V$) and Rest. The observations might be resulted from bad recording conditions.

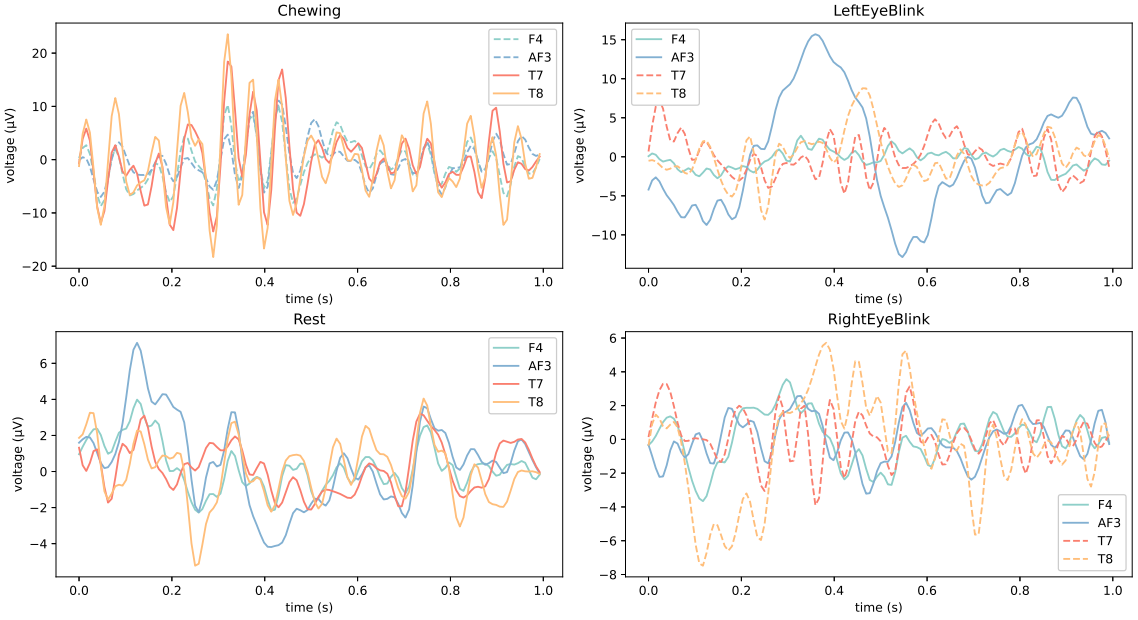

Fig. 6: Averaged ERP responses of four different channels (T7, AF3, F4, T8) under 4 events (Chewing, LeftEyeBlink, RightEyeBlink and Rest).

In order to further examine the validity of the extracted features, the corresponding features of every channel under every event were calculated. The results were displayed in **Fig. 7**, where the mean band power as well as respective standard deviation were depicted.

For the first feature (band power of 1-4 Hz), the band power of four channels at resting state were roughly 20 μV. Upon LeftEyeBlink, the band power of AF3 greatly increased to 65 μV, showing large difference with other three channels. Upon RightEyeBlink, the band power of F4 (aroung 25

μV) was only slightly larger than AF3 (20 μV). This was due to bad recording of RightEyeBlink data. Not as expected, the band power of 4 channels under Chewing events didn't experience notable increase. The results showed that eye blinking would indeed induce an increase in the power band 1-4 Hz, and also, the AF3 and F4 exhibited different response intensities upon the left and right eye blinking, which proved the effectiveness of band power of 1-4 Hz as a feature to identify eye blinks.

For the second feature (band power of 13-30 Hz), the band power of four channels at resting state was around 10 μV. It could be observed that all four channels would undergo an obvious increase in the band power, but clearly, T7 and T8 possessed more alteration. The similar behavior of F4 and AF3, T7 and T8 corresponded to the symmetry of the two pairs of channels. Moreover, it could be noticed that the band power of T7 and T8 would suffer a small increase under LeftEyeBlink and RightEyeBlink respectively, because the motion of the eyes might unavoidably affect T7 and T8 as well. The results showed that chewing would significantly affect the band power of T7 and T8 uniformly, while eye blinking would not result in a large increase of delta band power, which proved the effectiveness of band power of 13-30 Hz as a feature to identify chewing.

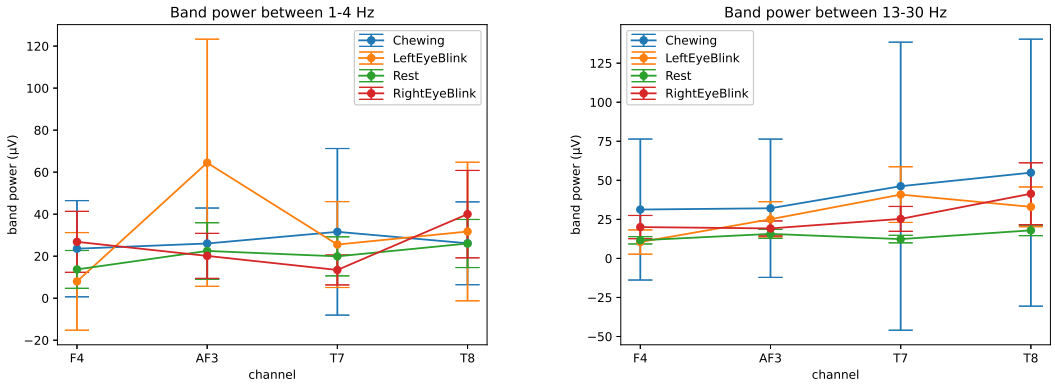

Fig. 7: Band power of relevant channel at different events. Left: Band power (1-4 Hz) of AF3, F4 ,T7 and T8 under respective four events. Right: Band power (13-30 Hz) of AF3, F4 ,T7 and T8 under respective four events.

The validity of two band powers as features was examined. Hence, the proper choice of features for eye blinking and chewing was as follows:

- **Eye Blinking:** Band power 1-4 Hz of AF3 and F4.
- **Chewing:** Band power 13-30 Hz of T7 and T8.

The feature vector of every time window was four dimensional. A simple and intuitive option is to train a machine learning model to classify the four cases. Therefore, Support Vector Machine (SVM) was trained with a random train-test split. 30-50 trials of every event formed the whole dataset. The test took up 20% of the dataset after shuffling. The confusion matrix of the trained SVM model on test samples was depicted in **Fig. 8 Left**. It could be observed that the accuracies of Rest, LeftEye and RightEye were satisfying and high (all above 80% correctness). However, the successfully detected Chewing percentage was relatively low (40%). This misclassification was due to poor quality of recording, especially the bad recording of RightEyeBlink.

Considering the bad classification of Chewing, the training excluding Chewing with the same SVM procedure was performed as well. The confusion matrix in this case was shown in **Fig. 8 Right**. After the exclusion of Chewing, the accuracy of other 3 cases was enhanced and was extremely high (over 90%). Therefore, SVM model without chewing might be a better option.

According to previous discussion, three classification methods were proposed:

- **a) Threshold:** Different thresholds for different events were used. The exact values of threshold were obtained empirically.
  - RightEyeBlink: Band power (1-4 Hz) of F4 channel larger than 60.
  - LeftEyeBlink: Band power (1-4 Hz) of F4 channel larger than 150.
  - Chewing: The mean of band power (13-30 Hz) of T7 and T8 channels larger than 200.

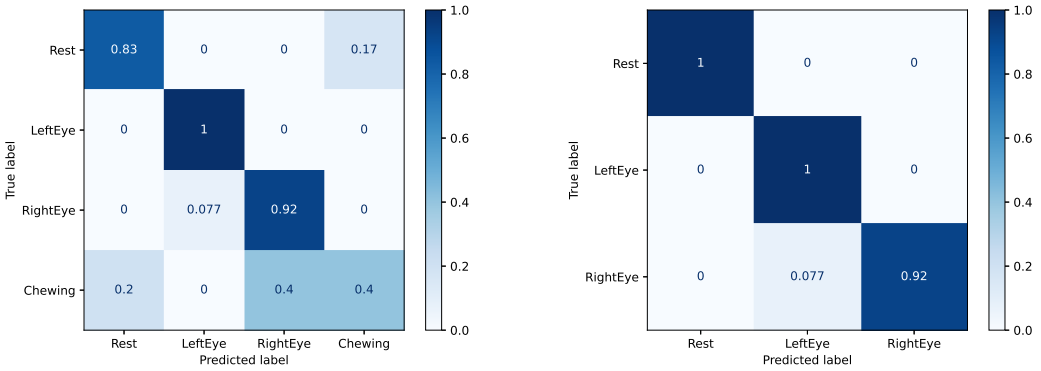

Fig. 8: ConfusionMatrix of SVM with training data. Left: Full training of SVM. Right: Training excluding Chewing

- **b) SVM:** The trained SVM was saved and used in real-time classification.
- **c) Mix:** The chewing was detected by threshold method and the other three cases were classified by trained SVM.

The above mentioned three methods were used in testing procedure, where a random sequence of RightEyeBlink, LeftEyeBlink, Chewing and Rest was produced. The sequence contained a total of 30 entries so that the duration of the whole experiment wouldn't last too long. The pilot would accomplish corresponding task when the pilot saw a sign on the computer, and the EEG signals were recorded. The recorded signals were processed in the same pipeline and classified using the three different methods. The accuracy of 3 methods were shown in **Tab. 1**.

method	Threshold	SVM	Mix
acc.	0.25	0.25	0.28

Table 1: Accuracy in different classification methods

Unfortunately, the test accuracy of all methods was unsatisfying ($\approx 25\%$). After implementing thorough investigation, possible causes for the bad performance were due to large variance of EEG signals from different sessions. Depending on the wetness and location of the electrodes, surrounding environment, mood of the pilot, all these factors might lead to various levels of EEG signals. Therefore, the threshold and the trained SVM model might malfunction if the current signal differs significantly from the training data.

To ensure that the training data really reflected the corresponding motions, Independent Component Analysis (ICA) was implemented on the training data. Given the typical features of eye blinks and chewing, the corresponding Independent Component (IC) was selected out and shown in **Fig. 9**. It could be verified from **Fig. 9 Left** that the training data indeed reflected the activity at the corresponding areas. However, as it could be observed in **Fig. 9 Right**, the response in every epoch/trial was asynchronous due to different reaction time and therefore might lead to inaccurate training of SVM model as well as threshold.

Due to the poor performance of these three methods, they needed to be adapted before the actual trial began. Therefore, the threshold method was selected since it demanded less effort to adapt, while the SVM method required offline training.

4 Discussions

Given the above discussion, the methods of detecting artifacts were easy to implement and possessed solid technical backgrounds. However, the following drawbacks and limitations could be concluded:

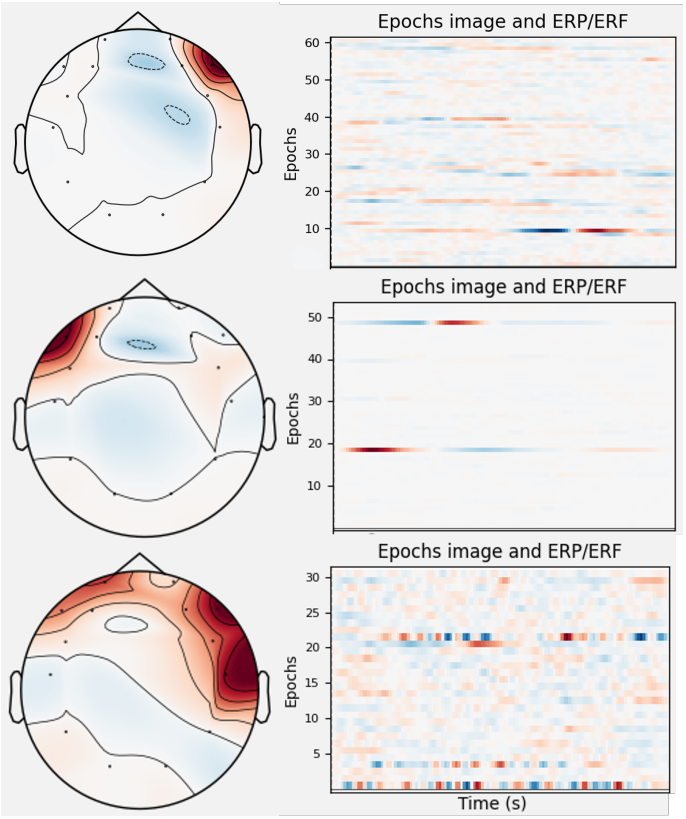

Fig. 9: Left: Topoplots of ICs under LeftEyeBlink, RightEyeBlink and Chewing respectively. Right: The development of activation of different epochs.

From the hardware perspective, our headset has a broken AF4 electrode, which located in anterior frontal lobe. AF4 sensor is prone to detect more features for right eye blink. Accordingly, when the AF4 electrode works, classification performance would be better.

The variance of EEG signals across experiment sessions was unexpectedly large, leading to difficulty in training machine learning model as well as other classification techniques such as thresholding. Therefore, standard conditions like wetness, exact positions of electrodes should be further settled down.

Since the individual differences in persons are large, trained classification model may not perform well for others/could not be shared across persons. In our case, it would be better for new users to collect new labeled data to train a new personal model. But this process can be time-wasting. To address this issue, developing a general cross-subject model could be a promising direction for future work.

Further optimization of the project could be implemented in the following aspects: Introduce more promising feature extraction methods such as Power Spectral Density and Template Matching [1]. The thresholding method could be improved by introducing adaptive threshold.

To simplify, we collected offline training data not in a random way, which caused Neural adaptation or sensory adaptation. Because when the nervous system is stimulated by repeated or or sustained events, the response to the stimulation will gradually decreases. To improvement, trials in one block should be conducted by random indications.

References

[1] Mohit Agarwal and Raghupathy Sivakumar. “Blink: A fully automated unsupervised algorithm for eye-blink detection in eeg signals”. In: *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2019, pp. 1113–1121.

[2] Andrew P Allen, Tim JC Jacob, and Andrew P Smith. “Effects and after-effects of chewing gum on vigilance, heart rate, EEG and mood”. In: *Physiology & Behavior* 133 (2014), pp. 244–251.

[3] Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. “Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis”. In: *Neuroimage* 34.4 (2007), pp. 1443–1449.

[4] Qiyun Huang et al. “An EOG-based human–machine interface for wheelchair control”. In: *IEEE transactions on biomedical engineering* 65.9 (2017), pp. 2023–2032.

[5] Alberto López et al. “Development of an EOG-based system to control a serious game”. In: *Measurement* 127 (2018), pp. 481–488.

[6] Dang-Khoa Tran, Thanh-Hai Nguyen, and Thanh-Nghia Nguyen. “Detection of eeg-based eye-blinks using a thresholding algorithm”. In: *European Journal of Engineering and Technology Research* 6.4 (2021), pp. 6–12.