SoftwareX 22 (2023) 101371

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

MartMi-BCI: A matlab-based real-time motor imagery brain-computer interface platform

Guoyang Liu [a,b,*], Janet H. Hsiao [a,c,d], Weidong Zhou [b], Lan Tian [b]

[a] Department of Psychology, The University of Hong Kong, Hong Kong SAR, China
[b] School of Microelectronics, Shandong University, 27 Shanda South Road, Jinan, Shandong, China
[c] The State Key Laboratory of Brain and Cognitive Sciences, The University of Hong Kong, Hong Kong SAR, China
[d] The Institute of Data Science, The University of Hong Kong, Hong Kong SAR, China

ARTICLE INFO

Article history:
Received 21 January 2023
Received in revised form 2 March 2023
Accepted 13 March 2023

Keywords:
Brain-computer interface (BCI)
Motor imagery
OpenBCI
Matlab
Real-time

ABSTRACT

Motor imagery brain-computer interface (MI-BCI) is a promising tool for neuro-rehabilitation. The real-time MI-BCI enables people with motor dysfunction disease to interact with the outside world. This work develops a Matlab-based real-time MI-BCI (MartMi-BCI) software, which involves two main modules, a real-time EEG analysis platform (RTEEGAP) and a model training platform (MTP). The RTEEGAP can realize real-time EEG analysis in time, frequency, and spatial domains and perform MI experiments with real-time feedback based on the OpenBCI device. Simultaneously, the MTP can train the CSP-based MI classification model and visualize the time–frequency reaction map. We hope the MartMi-BCI software can promote the development of EEG-based MI paradigm design and EEG classification algorithms.

Code metadata

Current code version	V1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00016
Permanent link to reproducible capsule	
Legal code license	GNU General Public License v3.0
Code versioning system used	GitHub
Software code languages, tools and services used	MATLAB
Compilation requirements, operating environments and dependencies	MATLAB R2019a and later releases, Deep Learning Toolbox, Signal Processing Toolbox, Statistics and Machine Learning Toolbox, Parallel Computing Toolbox (optional).
If available, link to developer documentation/manual	None
Support email for questions	gyangliu@hku.hk

1. Motivation and significance

With the development of brain computer interface (BCI), scalp electroencephalography (EEG) has become a popular tool for studying brain diseases and functions. And as a well-known exemplification of BCI interaction, motor imagery (MI) has been widely used to get the intention from the subjective [1,2]. The EEG-based MI-BCI can be applied both in biomedical fields such as the control of wheelchairs [3] or mechanical arms [4] and rehabilitation systems [5], as well as non-biomedical fields, such as entertainment and gaming [6,7].

Currently, many different types of methods are used in MI paradigm. Common Spatial Pattern (CSP) [8] is one of the most effective methods for feature extraction. It acquires the most representative features by means of the spatial filters in every two different classes. And based on CSP, Sub-Band CSP (SBCSP) [9], which added the frequency band division procedure and Filter Bank CSP (FBCSP) [10,11] with feature selection procedure were proposed to boost the performance of MI tasks. In addition to the CSP algorithm, some works also attempt to combine temporal and frequency features [12,13]. Riemannian approach [14] was another effective approach to extracting EEG-based MI features. However, the feature dimension will be significantly increased

[*] Corresponding author.
E-mail addresses: gyangliu@hku.hk (Guoyang Liu), jhsiao@hku.hk (Janet H. Hsiao), wdzhou@sdu.edu.cn (Weidong Zhou).

https://doi.org/10.1016/j.softx.2023.101371
2352-7110/

when the EEG channel rises. Although EEG-based MI-BCI has been researched for many years, there are still many problems to be solved. One of the critical issues is the low classification accuracy. To improve the accuracy, we adopted our previous work (an algorithm combined with divergence-based CSP with SVM) [15] to build a model training platform, achieving a high-performance BCI system. Simultaneously, another key problem is the optimum design of MI paradigm. A software platform that can easily design and testing the paradigm of MI task with feedback is urgently needed.

OpenBCI, as a popular open-source BCI platform, has been used in a variety of experiments to record the EEG signals, such as rehabilitation robot [16], P300 analysis [17], SSVEP analysis [16,17], MI tasks [17–19] and Emotion recognition task [20]. Compared with other BCI devices, it has a lower price and open-source hardware and GUI software. Meanwhile, it can also be compatible with standard EEG electrodes. Therefore, it is very convenient for researchers to conduct the experiment. Although the website of OpenBCI offers the software application programmed by Processing language to collect EEG data, there is no unified open-source platform for OpenBCI that can be directly used to design the MI paradigm with real-time feedback and training-efficient MI algorithms that can visualize the time–frequency brain activation.

This paper introduces a Matlab-based real-time MI-BCI (MartMi-BCI), which has complete MI paradigm functions, including extracting EEG data, preprocessing data, analyzing data in temporal and spatial fields, training models, and giving real-time feedback. And the platform is easy for people of all levels to operate because it uses a friendly graphical interface and it supports an open-source OpenBCI device. At the same time, the platform can also visualize different person's temporal-frequency reaction maps (TFRM), enabling the platform to enhance the interpretability of the model.

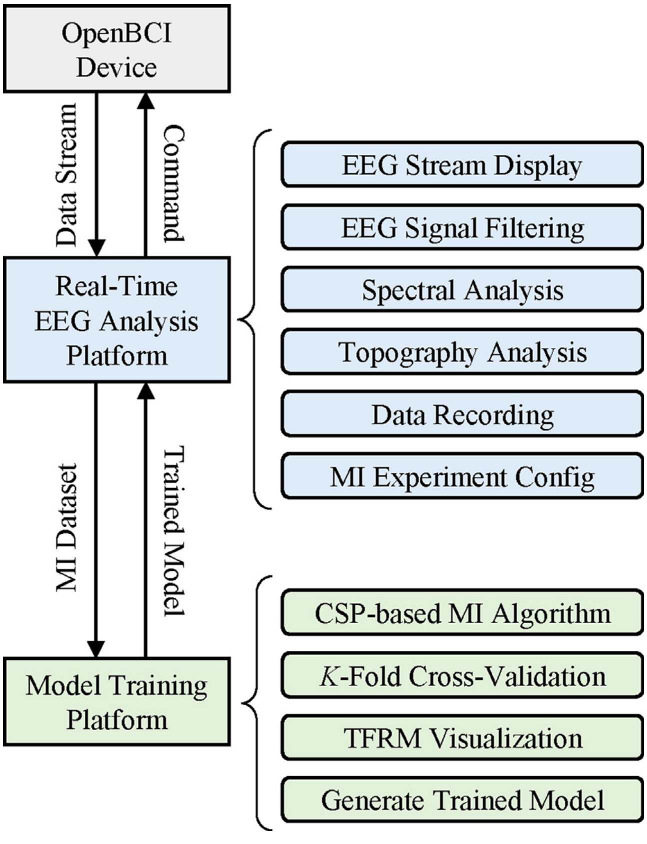


Fig. 1. The overview of the MartMi-BCI architecture.

2. Software description

2.1. Software architecture

Fig. 1 demonstrates the architecture of the proposed MartMi-BCI software. The MartMi-BCI contains two major platforms: a real-time EEG analysis platform (RTEEGAP) and a model training platform (MTP). The RTEEGAP supports multi-channel EEG stream display, EEG signal filtering using FIR and IIR filters, spectral analysis with the FFT algorithm, topography analysis, data recording, and real-time motor imagery experiments. The MTP can evaluate the quality of the motor imagery data with CSP-based algorithms and visualize the TFRM of the subject. Meanwhile, it provides the RTEEGAP with a well-trained motor imagery model to achieve real-time motor imagery tasks.

2.2. Software functionalities

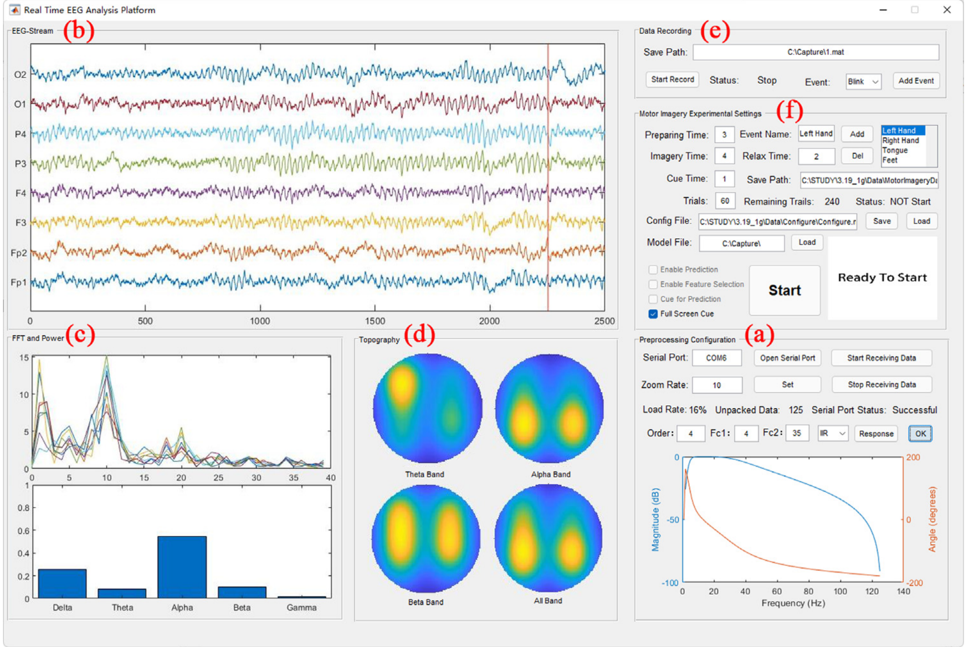
2.2.1. Real-time EEG analysis platform (RTEEGAP)

2.2.1.1. Preprocessing configuration. As shown in Fig. 2(a), the serial port setting, display setting, and filtering setting can be operated on the preprocessing configuration panel. When entering the RTEEGAP, we need to set the serial port correctly to connect the OpenBCI device. Firstly, enter the serial number of the OpenBCI device, and then click the "Open Serial Port" button and "Start Receiving Data" in turn. If the device is connected successfully, the "Serial Port Status" will display "Successful". Simultaneously, the "Load Rate" displays the computational load of the RTEEGAP software. If the load rate exceeds 90%, it indicates that the performance of the current computer is relatively low, which may affect the real-time display of the software. The "Unpacked Data" demonstrates the number of samples received in one refresh interval (the default refresh interval is set to 0.5s). Besides, we can adjust the amplitude of the EEG stream displayed in Fig. 2(b) by setting the "Zoom Rate". Note that RTEEGAP software uses the function 'decodeOpenBCIData' to decode the OpenBCI serial data. Users can modify this function to make this software compatible with other EEG recording systems.

On the other hand, we can define different filtering settings in this panel. Two types of band-pass filters, namely the Finite Impulse Response (FIR) filter, and Infinite Impulse Response (IIR) filter, are supported for preprocessing the raw signals. When the order and cut-off frequency (Fc1 and Fc2) are entered, we can click the "Response" button to check the magnitude–frequency and phase-frequency responses of the filter. Then, click "OK" to apply the set filter to the raw signal.

2.2.1.2. EEG stream display. Once the software starts receiving data, the preprocessed multi-channel EEG stream is displayed in Fig. 2(b). The default refresh interval is set to 0.5s, and it can be modified by changing the "DispMs" field value in the global variable "IntervalTime". At the same time, the default display length and sampling rate are set to 10 s, and they can be adjusted in the global variable "DispSec" and "SampleRate".

2.2.1.3. Spectral analysis. Fig. 2(c) shows the FFT and power spectral analysis. The x-axis values of the FFT figure represent the frequency in Hz, while the y-axis values of the FFT figure represent the relative amplitude. Each line corresponds to one channel, and the color matches the color of the EEG stream displayed in Fig. 2(b). Meanwhile, the normalized power spectral over all channels is presented in the bottom figure, considering five typical EEG bands, including Delta (1–3 Hz), Theta (4–7 Hz), Alpha (8–13 Hz), Beta (14–30 Hz), and Gamma (31–45 Hz) bands.


Real Time EEG Analysis Platform

Data Recording (e)

Save Path: C:\Capture\1.mat

Status: Stop

Event: Blink

Motor Imagery Experimental Settings (f)

Preparing Time: 3
Event Name: Left Hand
Imagery Time: 4
Relax Time: 2
Cue Time: 1
Save Path: C:\STUDY\3.19_1g\Data\MotorImageryDa
Trials: 60
Remaining Trails: 240
Status: NOT Start
Config File: C:\STUDY\3.19_1g\Data\Configure\Configure.r
Model File: C:\Capture\

□ Enable Prediction
□ Enable Feature Selection
□ Cue for Prediction
☑ Full Screen Cue

Ready To Start

Preprocessing Configuration (a)

Serial Port: COM6
Zoom Rate: 10
Load Rate: 16% Unpacked Data: 125 Serial Port Status: Successful
Order: 4 Fc1: 4 Fc2: 35 IIR

Fig. 2. The Matlab-based GUI of real-time EEG analysis platform. The panel includes following parts: (a) preprocessing configuration, (b) EEG-stream display, (c) FFT and power display, (d) topography analysis, (e) data recording, (f) motor imagery experimental settings.

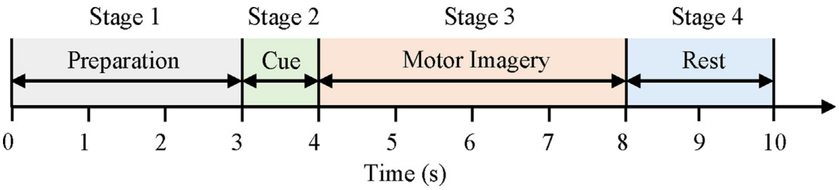


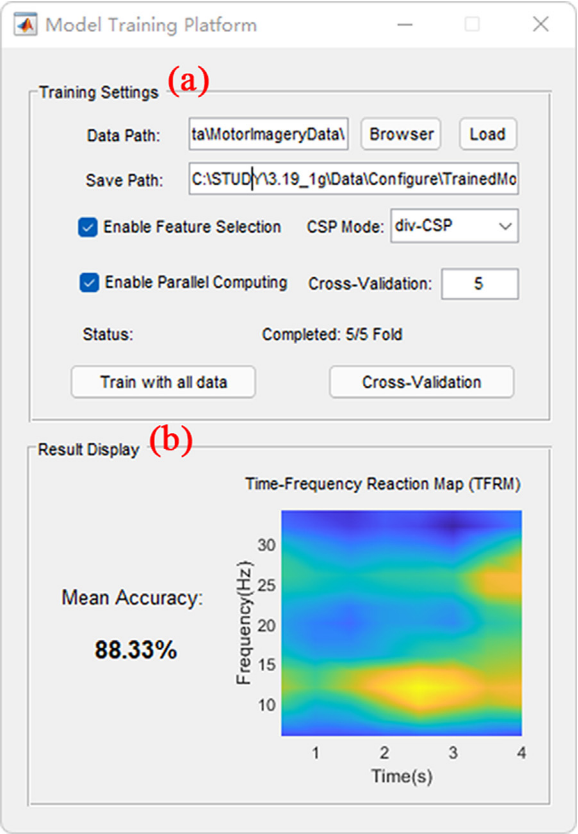
Fig. 3. The four-stage workflow of a motor imagery trial.

2.2.1.4. Topography. The EEG topography visualizes the activated brain regions by computing the spectral power value of each channel. In Fig. 2(d), the topography of three sub-bands (Theta, Alpha, Beta) and all EEG bands is illustrated. The function of plotting topography is based on the convolution operator. First, mark the coordinates of each channel on a map matrix of size 128 × 128 (corresponding to a circular area with a radius of 64) according to the actual channel position. These coordinates can be set in "x" and "y" field values in the global variable "ElectrodeMap". Then, the weights of all channels are computed and stored at the corresponding coordinate positions in the matrix. Finally, the map matrix is convoluted by a 128 × 128 Gaussian kernel with a standard deviation setting of 25, and the topography is obtained.

2.2.1.5. EEG data recording. When the software is receiving data, we can use this panel to record the received data and mark the event. We can click the "Start Record" to start the EEG recording. During EEG recording, the event can be selected and added to the record file. The record mat file consists of one structure variable named "RecordData", where the "NowData" field matrix stores the recorded preprocessed EEG data, and the "Time" and "Sample" represent the total recorded time (seconds) and samples, respectively.

2.2.1.6. Motor imagery experimental settings. The motor imagery experimental setting panel can customize different motor imagery paradigms. Usually, the motor imagery paradigm can be divided into four stages, i.e., preparing stage, cue stage, imagery stage, and relax stage. Fig. 3 depicts an example of a classic motor imagery paradigm. In this software, the time of each stage can be set in the corresponding text box. Meanwhile, the motor imagery task can be edited in the corresponding list. The default motor imagery task list contains four typical motor imagery tasks: Left Hand, Right Hand, Tongue, and Feet. The illustrative pictures are stored in "\Data\Picture" folder. The number of trials of each task is determined in the "Trials" text box. By editing the "Save Path" text box, the save path of the record motor imagery data file can be defined. All above settings are able to be saved as a config file, and its save path is defined in the "Config File" text box. When the program is next started, the config file can be loaded directly to set the paradigm. After setting the parameters of the paradigm properly, the MI experiment can be started trial-by-trial by clicking the "Start" button. If the "Full Screen Cue" is enabled, the cue picture will be displayed in full screen of the software, or else the cue picture will only be displayed in the right small area of the panel "Motor imagery experimental settings".

In addition, the software also supports real-time MI feedback to verify the performance of the MI algorithm once the MI classification model was trained and generated with MTP. Detailly, click the "load" button to select a trained model file, and then the "Enable Prediction", "Enable Feature Selection", and "Cue for Prediction" checkboxes will be available. Check the "Enable Prediction" check box to execute the motor imagery paradigm with real-time feedback. If the trained model contains feature selection



Training Settings

Data Path: ta\MotorImageryData\

Save Path: C:\STUDY\3.19_1g\Data\Configure\TrainedMo

☑ Enable Feature Selection CSP Mode: div-CSP

☑ Enable Parallel Computing Cross-Validation: 5

Status: Completed: 5/5 Fold

Result Display

Mean Accuracy:

88.33%

Fig. 4. The Matlab-based GUI of model training platform. The panel consists of the following parts: (a) training settings, (b) result display.

information, then the "Enable Feature Selection" can be checked to improve inference efficiency. With the default settings, no cue is provided when the paradigm has feedback, and the cue picture is replaced with a fixed cross picture (see Fig. 6(b)). However, if the "Cue for Prediction" is checked, then the cue will be retained to validate the consistency of the cues and feedback. According to our previous work [15], the latency of the real-time feedback caused by model computation will be less than 15 ms when processing 8-channel data (with MATLAB 2019a platform running on a desktop with 4.0 GHz i7-8700 processor and 32G RAM.), which can fully meet the requirement of real-time feedback.

When each trial is completed, the relevant data are saved in the "Data\MotorImageryData" folder under the root directory of the software, and the file name is the trial number. Each trial file contains one structure variable named "MotorImagery". The main field names are "RecordData", "RecordRawData", "Sample", and "ClassOrder". Among them, the "RecordData" and "RecordRawData" matrix store the filtered and non-filtered EEG data of one trial. The "Sample" is also a structure variable, where the field values "Stage 1" to "Stage 4" provide the last sample index of each stage, and the field value "Cnt" provides the total number of recorded samples for the current trial. The "ClassOrder" vector stores the class label of all trials, and the label of the current trial can be found by indexing the trial number.

2.2.2. Model training platform (MTP)

2.2.2.1. Training settings. As shown in Fig. 4, In the training settings panel, various training hyper-parameters can be set. First, click the "Brower" button to select the folder containing training data (the default path is "Data\MotorImageryData"). Then, click the "Load" button to verify and load the training data to the workspace. Two CSP modes, namely the traditional CSP method [21] and the div-CSP method [22], are available for feature extraction. A wrapper-based feature selection algorithm will be applied if the "Enable Feature Selection" is checked. To accelerate the training procedure, the "Enable Parallel Computing" can be checked. With this setting, the software will use the "parfor" command to enable parallel computing using Matlab parallel pool. The training scheme is the same as our previous work [15]. In brief, the raw EEG signals are first segmented into 297 Time–Frequency Segment (TFS) according to a certain scheme. Then all the TFS features are concatenated and fed into the SVM with linear kernel for classification. With wrapper-based feature selection, half a number of the TFS features can be discarded, improving the efficiency and accuracy of the model. For the technical details of the CSP/divCSP feature extraction module and of the classifier, please refer to our previous publication [15].

The MTP supports two training modes: "Cross-Validation" mode and "Train with all data" mode. The cross-validation mode is used to evaluate the MI classification performance of the subjects and visualize their Time–Frequency Reaction Map (TFRM). We can define the number of folds in cross-validation by editing the "Cross-Validation" text box. Notably, the greater the fold number, the more accurate TFRM and mean accuracy will be estimated. On the other hand, the "Train with all data" mode takes all the data as training data, and a model file will be generated after training. The model file contains a structure variable "ParaImagery". In its field variables, the "Wcsp" stores all of the trained CSP projection matrices, the "accMat" stores TFS scores, the binary variable in "PatCspIdx" stores the indexes of the selected TFS, the "Trained_LSVM" is the SVM classifier with linear kernel.

2.2.2.2. Result display. The result display panel displays the cross-validation results in the left region and visualizes the TFRM in the right area. The visualization of TFRM is achieved by averaging the TFS scores in corresponding time and frequency bands. The detailed procedure is provided in [15]. By visualizing the TFRM, the time–frequency reaction characteristic of the subject can be obtained and further analyzed.

3. Illustrative examples

Fig. 2 shows an example of the running software. First, the "Serial Port" is set to "COM6", and then click the "Open Serial Port" and "Start Receiving Data" Buttons in turn to connect the OpenBCI device. Here, we can see that the "Load Rate" is 16%, and the "Unpacked Data" is 125 (with the default refresh interval setting to 0.5s), which means that the sampling rate is 250 Hz. A 4-order band-pass IIR filter with cut-off frequencies between 4 Hz and 35 Hz is applied to 8-channel EEG data for preprocessing. Fig. 2(b) illustrates an example EEG stream with the subject's eye closed. It can be seen that the EEG stream contains quantities of rhythmic waves. From Fig. 2(c), we can observe that the 10 Hz frequency component (Alpha band) is significantly different from other frequency components, which matches the conclusion of the previous studies [23]. The topography shown in Fig. 2(d) indicates that the Alpha band in the occipital region is activated. This phenomenon is also observed in many previous studies [24,25].

As conducting the motor imagery experiment, Fig. 5(a)–(d) present an example of the four stages of one trial, which enables the "Full Screen Cue". At the end of each trial, a warning box will be appeared to check if the user wants to save this trial. If the displayed EEG data contain obvious noises, the user can select "No" to discard this trial. When all the trials are conducted, we can open the MTP software to train the obtained MI-EEG data.

As illustrated in Fig. 4(a), the "Data Path" is first set to the path filled in the "Save Path" of Fig. 2(f), and click the "Load"

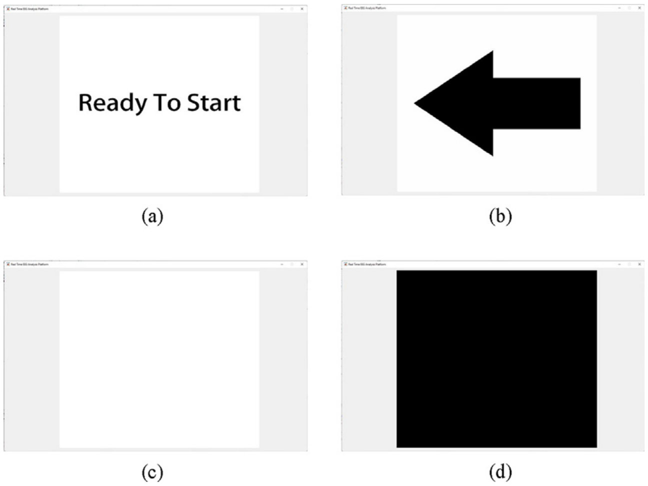


Fig. 5. An example of the motor imagery paradigm. (a)–(d) illustrate the panel display of the RTEEGAP from stage 1 to stage 4, respectively.

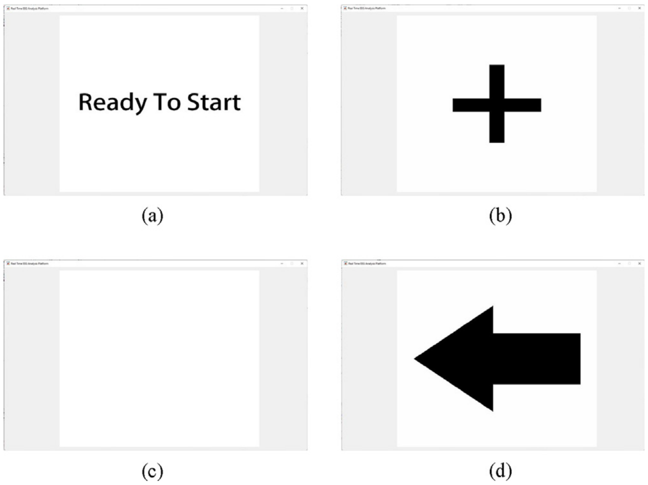


Fig. 6. An example of the motor imagery prediction paradigm. (a)–(d) illustrate the panel display of the RTEEGAP from stage 1 to stage 4, respectively.

button to verify and load all the training data. Next, the "CSP Mode" is selected as "div-CSP", and the feature selection and parallel computing function are enabled. After clicking the "Cross-Validation" button, we can wait for the algorithm to finish. The command window of Matlab will display the executing progress of the algorithm. When the algorithm is finished, the results are displayed in Fig. 4(b). It can be seen that the experimental subject achieves a mean accuracy of 88.33% in the 5-Fold cross-validation scheme. Simultaneously, the right region in Fig. 4(b) demonstrates the TFRM of the subject. We can observe that the 10-15 Hz frequency band in 2-3s is most activated, and a high-frequency band of around 25 Hz is also activated. However, the activated high-frequency band appears in the last second, which indicates that the subject could enhance the performance of executing motor imagery tasks by increasing the time of motor imagery. Lastly, if obtained mean accuracy matches the real-time feedback requirement, the "Train with all data" button can be pressed to generate the motor imagery model.

The RTEEGAP supports real-time motor imagery feedback. In detail, click the "Load" button to select the trained model file and enable the "Enable Prediction" check box. After that, click the "Start" button to execute the motor imagery task with real-time feedback. As shown in Fig. 6, a four-stage paradigm is conducted, where the last stage presents the predicted motor imagery task.

4. Impact

With the MartMi-BCI, users can easily design proper MI paradigms and conduct EEG-based MI experiments using the RTEEGAP module. The time–frequency-spatial EEG analysis and real-time feedback are supported. Users can also collect their own EEG databases with markers by RTEEGAP. Meanwhile, the MTP helps users automatically train the high-performance model of multiclass MI classification introduced in our published work [15]. It is suggested that the classification accuracy and computational complexity of the model outperform traditional MI-BCI algorithms [23,26,27]. Benefiting from the high efficiency of the MI classification model, the MartMi-BCI can achieve excellent real-time performance. The whole software system is implemented in GUIDE of Matlab, and the hardware is based on Open-BCI, which is user-friendly for new researchers and easy to be modified and customized by skilled researchers.

5. Conclusion

A user-friendly MI-BCI platform, MartMi-BCI, is established to let users of all levels achieve their EEG-related studies more conveniently and efficiently. Two modules, namely RTEEGAP and MTP, are involved in MartMi-BCI. RTEEGAP enables users to design MI paradigms, conduct efficient MI experiments, and analyze EEG signals, while MTP provides a user-friendly platform to train high-performance MI classification models. Currently, the software supports the connection with an open-source OpenBCI device and a CSP-based EEG classification algorithm. In future works, we will continue to update this software to support more EEG recording devices and more EEG classification algorithms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The support in part by the Independent Innovation Project of Shanliang Artificial Intelligence Research Center of Shandong University (No. 2018ST01) and the Key Program of Natural Science Foundation of Shandong Province (No. ZR2020LZH009). We also acknowledge Rui Zhang and Chaosong Li from Shandong University for the help in this work.

References

[1] Pfurtscheller G, Neuper C, Flotzinger D, Pregenzer M. EEG-based discrimination between imagination of right and left hand movement. Electroencephalogr Clin Neurophysiol 1997;103(6):642.

[2] Pfurtscheller G, Brunner C, SchlöGl A, Silva F. Mu rhythm (de)synchronization and EEG single-trial classification of different motor imagery tasks. Neuroimage 2006;31(1):153–9.

[3] Pinheiro OR, Alves LRG, Souza JRD. EEG signals classification: Motor imagery for driving an intelligent wheelchair. IEEE Lat Am Trans 2018;16(1):254–9. http://dx.doi.org/10.1109/TLA.2018.8291481.

[4] Edelman BJ, et al. Noninvasive neuroimaging enhances continuous neural tracking for robotic device control. Science Robotics 2019;4(31):eaaw6844.

[5] Mak A, Rd A, Hki B, Sp A. Review on motor imagery based BCI systems for upper limb post-stroke neurorehabilitation: From designing to application, Comput Biol Med 123.

[6] Plass-Oude Bos D, et al. Brain-computer interfacing and games. In: Tan DS, Nijholt A, editors. Brain-computer interfaces: applying our minds to human-computer interaction. London: Springer London; 2010, p. 149–78.

[7] Padfield N, Zabalza J, Zhao H, Masero V, Ren J. EEG-based brain-computer interfaces using motor-imagery: Techniques and challenges. Sensors 2019;19(6).

[8] Koles ZJ, Lazar MS, Zhou SZ. Spatial patterns underlying population differences in the background EEG. Brain Topogr 1990;2(4):275–84. http://dx.doi.org/10.1007/BF01129656.

[9] Novi Q, Guan C, Dat TH, Xue P. Sub-band common spatial pattern (SBCSP) for brain-computer interface. In: 2007 3rd international IEEE/EMBS conference on neural engineering. 2007, p. 204–7. http://dx.doi.org/10.1109/CNE.2007.369647.

[10] Ang KK, Chin ZY, Zhang H, Guan C. Filter bank common spatial pattern (FBCSP) in brain-computer interface. In: Proceedings of the international joint conference on neural networks. 2008, p. 2390–7. http://dx.doi.org/10.1109/IJCNN.2008.4634130, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-56349090127&doi=10.1109%2fIJCNN.2008.4634130&partnerID=40&md5=b6a886b233ce9ba1468241b210d88fcd.

[11] Ang KK, Chin ZY, Wang C, Guan C, Zhang H. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. Front Neurosci 2012;6:39. http://dx.doi.org/10.3389/fnins.2012.00039.

[12] Miao M, Zeng H, Wang A, Zhao C, Liu F. Discriminative spatial-frequency-temporal feature extraction and classification of motor imagery EEG: An sparse regression and Weighted Naïve Bayesian Classifier-based approach. J Neurosci Methods 2017;278:13–24. http://dx.doi.org/10.1016/j.jneumeth.2016.12.010.

[13] Miao Y, et al. Learning common time-frequency-spatial patterns for motor imagery classification. IEEE Trans Neural Syst Rehabil Eng 2021;29:699–707. http://dx.doi.org/10.1109/TNSRE.2021.3071140.

[14] Yang P, Wang J, Zhao H, Li R. MLP with Riemannian covariance for motor imagery based EEG analysis. IEEE Access 2020;8:139974–82. http://dx.doi.org/10.1109/ACCESS.2020.3011969.

[15] Liu G, Tian L, Zhou W. Multiscale time-frequency method for multiclass motor imagery brain computer interface. Comput Biol Med 2022;143:105299.

[16] Perera CJ, Naotunna I, Sadaruwan C, Gopura R, Lalitharatne TD. SSVEP based BMI for a meal assistance robot. In: IEEE international conference on systems, 2016.

[17] Durka PJ, et al. User-centered design of brain-computer interfaces: OpenBCI.pl and BCI Appliance. Bull Pol Acad Sci Tech Sci 2012;60(3):427–31. http://dx.doi.org/10.2478/v10175-012-0054-1.

[18] Olivas-Padilla BE, Chacon-Murguia MI. Classification of multiple motor imagery using deep convolutional neural networks and spatial filters. Appl Soft Comput 2019;75:461–72. http://dx.doi.org/10.1016/j.asoc.2018.11.031.

[19] Sawangjai P, Hompoonsup S, Leelaarporn P, Kongwudhikunakorn S, Wilaiprasitporn T. Consumer grade EEG measuring sensors as research tools: A review. IEEE Sens J 2020;20(8):3996–4024. http://dx.doi.org/10.1109/jsen.2019.2962874.

[20] Lakhan P, et al. Consumer grade brain sensing for emotion recognition. IEEE Sens J 2019;19(21):9896–907. http://dx.doi.org/10.1109/jsen.2019.2928781.

[21] Lemm S, Blankertz B, Curio G, Muller K-R. Spatio-spectral filters for improving the classification of single trial EEG. IEEE Trans Biomed Eng 2005;52(9):1541–8.

[22] Samek W, Kawanabe M, Müller K-R. Divergence-based framework for common spatial patterns algorithms. IEEE Rev Biomed Eng 2013;7:50–72.

[23] Kirkup L, Searle A, Craig A, McIsaac P, Larsen G. Three methods compared for detecting the onset of alpha wave synchronization following eye closure. Physiol Meas 1998;19(2):213.

[24] Plotkin WB, Cohen R. Occipital alpha and the attributes of the alpha experience. Psychophysiology 1976;13(1):16–21.

[25] Hohaia W, Saurels BW, Johnston A, Yarrow K, Arnold DH. Occipital alpha-band brain waves when the eyes are closed are shaped by ongoing visual processes. Sci Rep 2022;12(1):1–10.

[26] Al-Saegh A, Dawwd SA, Abdul-Jabbar JM. CutCat: An augmentation method for EEG classification. Neural Netw 2021;141:433–43.

[27] Wang B, et al. Common spatial pattern reformulated for regularizations in brain–computer interfaces. IEEE Trans Cybern 2020;51(10):5008–20.