“BCI for everyone” A subject-independent approach to EEG data

Inês Jesus†, José Chacón‡

***Abstract*—Encephalogram (EEG) motor imagery (MI) is a key technique in brain-computer interfaces (BCIs), facilitating diverse applications such as assistive robots and smart healthcare solutions. This paper evaluates various configurations of Convolutional Neural Networks (CNNs) applied to a MI paradigm in a setting of subject-independence, an area that is underexplored in current literature. The focus on subject-independence aims to enhance the practicality of these models by improving their ability to generalize across different individuals, thereby broadening the real-world applicability of EEG-based MI systems.**

***Index Terms*—Electroencephalogram, Motor Imagery, Convolutional Neural Networks, Subject-Independence**

I. Introduction

Motor imagery (MI)-based brain-computer interfaces (BCIs) have very diverse applications in neuroscience and technology, offering significant potential to enhance the quality of life, particularly for individuals with motor and communication impairments. This paper focuses on the motor applications of electroencephalograms (EEGs), which provide a compelling alternative to invasive methods due to the ease of use, low cost, robustness and portability of EEG devices.

Recent advancements in deep learning have significantly enhanced the sophistication and capabilities of MI-EEG-based applications, allowing for better handling the complex nature of this data. In this study, we explore the effects of different deep learning approaches on a MI paradigm using a large EEG dataset [1].

Existing literature primarily employs a subject-dependent approach to training models [2], wherein data from a single subject is used for both training and evaluation. While this method yields high classification accuracy for the individual, it fails to generalize across different subjects.

To address this limitation, we contribute to subject-independent cross-subject research, utilizing data from multiple subjects for training and data from unseen subjects for evaluation. Our goal is to develop a model with sufficient generalization capability and robust performance, which can later be used for transfer learning across various subjects.

This report is structured as follows: Section II reviews the state of the art, followed by a technical description of the data models in Section III. Section IV presents the proposed signal pre-processing techniques. Section V delves into the learning model, detailing its mechanisms. Performance evaluation is carried out in Section VI, and concluding remarks are provided in Section VII.

†Department of Mathematics, University of Padova, email: ines.caeironevessilveirajesus@studenti.unipd.it

‡Department of Mathematics, University of Padova, email: jose-javier.chaconmejia@studenti.unipd.it

II. Related Work

The study in [3] focused on validating existing data augmentation approaches on EEG data. They performed data augmentation on two different datasets through the implementation of some time, frequency and spatial transformations. One of the performed methods was `BandstopFilter`, motivated by the fact that randomly filtering specific frequencies would prevent the models from overfitting on subject specific features and giving too much importance to narrow frequency regions. It can be seen as a type of dropout for frequencies. On a EEG dataset built on a similar paradigm to the one used in this study it was discovered that a filter with small bandstop width could be beneficial to BCI applications.

Other interesting data augmentation technique discussed in the same study is `ChannelsDropout`, which provided a significant performance boost. With the spread of mobile EEG devices, a big challenge is to create models that are robust to noise and missing values, such as entire channels. The research community is thus presenting great interest in increasing the transferability across datasets considering different number or order of channels. Therefore, `ChannelsDropout` is a technique that is expected to greatly promote the generalization of the model by not allowing it to rely to much in specific channels and instead learn general patterns across all of the channels.

Different types of deep models have been employed in the classification of EEG data, from discriminative models, to representative models, to generative models and to hybrid models.

Generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have been showing promising results to augment and improve training data. However, using a CNN provided us with a better comparison power with other studies, since there already does not exist an extensive literature in the subject-independency field, added with a wide variety of methods available for the implementation of such networks.

The Shallow and Deep ConvNets are proposed by [4]. They have achieved competitive results in recent years but have some disadvantages to it. They have many hyperparameters to optimize, they may require a large amount of training data and may take longer to train than simpler models. While the Deep ConvNet was designed for more general purposes, the

Shallow ConvNet is meant to be used on oscilatory data, so it may perform worse with other type of EEG problems.

The EEGNet [5] is a very robust network that gives competitive results in a wide range of EEG problems with very little complexity. In particular, in the cross-subject training technique, it can achieve as good results as the previously mentioned models, with an incredibly small fraction of the number of parameters. As such, this is an interesting model to consider when experimenting with different approaches.

III. Processing Pipeline

In this section, we explain the steps taken to achieve the final results.

We started by doing data preprocessing by applying some methods explored in literature. First, we do low-pass filtering and then we proceed to segment the recording session samples in trials of 210 data points. The dataset was split in training and testing, keeping three subjects for testing the models, since we want to understand how the models behave when applied to new subjects. These processes are further discussed in Sec. IV.

Every model architecture used is enriched with data augmentation layers. After data augmentation, there is also normalization of the batch according to previously extracted mean and variance from the training dataset of the current fold. These methods were implemented on the architecture of the model to significantly improve memory complexity, at a very low possible increase of cost on the time complexity.

Leave One Subject Out cross-validation is used to evaluate our models, so for every configuration of training and evaluation subjects (there are 10 subjects, so 10 folds) we train the model from scratch for 20 epochs and then take the mean accuracy across the 10 models.

After preforming the cross-validation, the model configuration that resulted in the best mean accuracy is used to train the model from the beginning using the 10 subjects previously selected for training.

The general setup of our experiments is represented in Fig. (reference).

IV. Signals and Features

The EEG motor imagery dataset is an open source dataset for BCI applications containing 60 hours of EEG recordings, across 13 participants and 75 recording sessions in 4 interaction paradigms. For this project, we chose the HaLT paradigm based on six mental states of left and right-hand movements, left and right-leg movements, tongue movement and a passive mental imagery in which participants engaged in no motor imagery.

In each trial, an action signal indicating one of the six states was presented for 1s during which the participants implemented the selected motor imagery once, except with the passive state (the circle represented in Fig. 2), for which they remained neutral. The participants then remained passive until the next action signal presentation, in variable intervals of 1.5-2.5s.

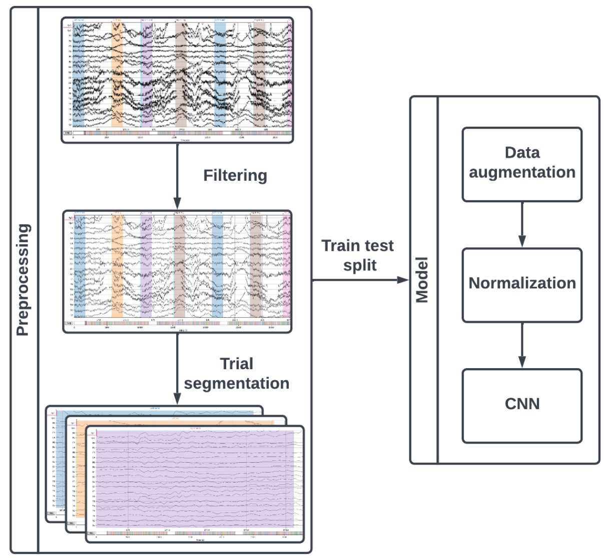


Fig. 1: Processing Pipeline diagram

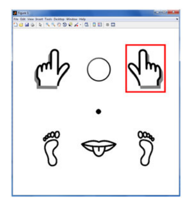


Fig. 2: Graphical user interface used for HaLT interactions

This trial procedure was repeated 300 times per interaction segment, for a total of three interaction segments per recording session.

The sampling frequency was regular for the whole dataset, at 200 Hz.

We performed minimal preprocessing to the data, since the Deep Learning models are supposed to capture the complex patterns in the data.

Frequency filtering is able to remove a large portion of noise, such as low frequency artifact and high frequency noise. We first applied to all recording sessions a causal third-order Butterworth filter to low-pass filter the signal frequencies below 38 Hz, as suggested by [2]. In [4], the data was also filtered with high pass with cutoff 4 to remove possible eye movement artifacts, but in the extensive literature analysis of [2] it is shown that considering the lower frequencies often improves model accuracy.

The next step was to segment the samples by trials. For this, some experiments were made to analyze what was the best segmenting strategy. The first approach was to consider the entire 2.5s after the action signal is presented to participants, with hopes that the 1.5s after the signal ended with still provide the models with some information.

In an attempt to improve our results, we resorted to some suggestions described in the literature.

The study in [4] started their trials 0.5s before the cue.

A second approach was the use of sliding time windows, so each trial was divided in three windows, as a way to increase the training size and to prevent the model from overfitting on phase information within the trial (the location and timing of points in the trial).

Using the entire trial showed poor results when starting 0.5s before the cue. Furthermore, the cropping strategy resulted in even worse results, so we further experimented with entire trials.

One last segmentation analysis was made by considering each trial as the entire duration of the action signal cue, which was somewhat variable (it was a little more than 1s at times). As such, samples of size 210 (1.05s) were cut for each trial, starting from the beginning of the cue. This strategy produced similar results to the first approach with 500 samples. As such, to improve time and memory complexity, this was the primary sample segmentation strategy used for later results.

For training, validation and testing, the dataset was split in the following way. Since we want to make our model robust to subject-independent applications, 3 participants were randomly selected to form the test dataset. The rest was kept for training and evaluation. During the training of the model we performed Leave One Subject Out (LOSO) cross-validation, which is the most implemented validation method in the literature for cross-subject experiments of EEG data [2], which means that we performed cross-validation iterating through all of the subjects to leave one out each time.

V. Learning Framework

The considered model was the EEGNet [5]. This was the chosen model because it is a CNN model with a light architecture and very few parameters to train, resulting in good memory and time complexity values. Its performance was evaluated in cross-subject contexts and it achieve competitive results with respect with other CNNs. Furthermore, opposite to some other CNN models, its was designed to be able to generalize across different BCI paradigms, without any restrictions to specific feature types. The architecture of the EEGNet, presented in Fig. 3 can be decomposed in four main stages.

- **Temporal convolution.** The model starts with a temporal convolution to learn frequency filters. It has a `F1` hyper-parameter that represents the number of 2D convolutional filters of size (1,`S`) where `S` should be set to half of the sampling frequency, in our case $S = 100$.
- **Depthwise convolution.** Then it proceeds to use a depth-wise convolution layer to learn frequency-specific spatial filters by tying spatial filters directly to a temporal filter. The size of the filter here is (`C`,1) where `C` is the number of channels. For this study, $C = 21$. It has a depth hyperparameter `D` that represents the number of spatial filters to learn within each temporal convolution. After this, Batch Normalization is applied along the feature map dimension followed by an activation function. The original activation function used is ELU but we also experiment with ReLU. This is followed by an average pooling layer to reduce the sampling rate of the signal. We treat the size of the pooling layer as a hyperparameter. Finally dropout is performed, with a dropout probability of *p*, which is recommended to use at 0.25 for cross-subject classification since the training set sizes are much larger.
- **Separable convolution.** The third stage consists of a combination of a depthwise convolution which learns a temporal summary for each feature map followed by a pointwise convolution which learns how to optimally mix the feature maps together. Here, a hyperparameter `F2` determines the number of pointwise convolutions. This is followed by another average pooling layer to further decrease the dimensions.
- **Classifier.** Finally it has a dense layer to perform classification of the signal. The features here are passed to a softmax classification with `N` units (number of classes to classify). For the paradigm considered, `N` is 6.

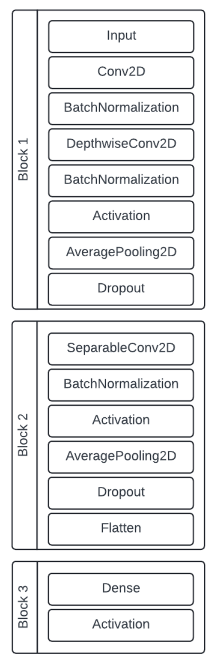


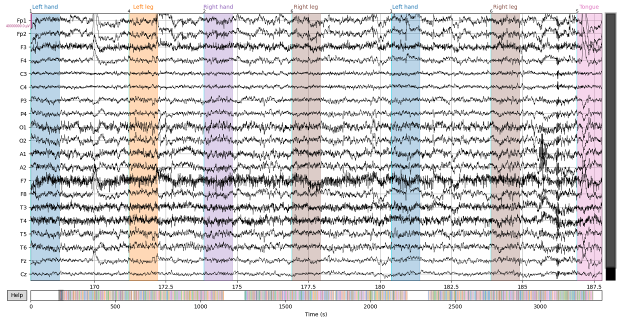
Fig. 3: EEGNet architecture

VI. Results

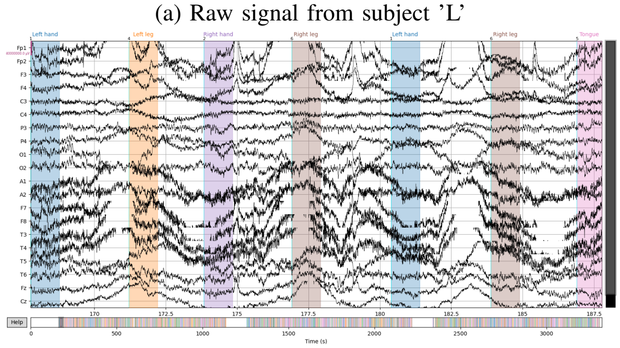
Something interesting that was evident while performing LOSO cross-validation was that some subjects when used for validation would give way below average results, while others gave a really good validation accuracy. This was the results of validation for the model X:

As we can see, the subjects X, Y and Z present a very low validation accuracy, which might indicate that their data is noisier than others. On the contrary, the model was accurate in predicting the action signal for subject W, meaning that...

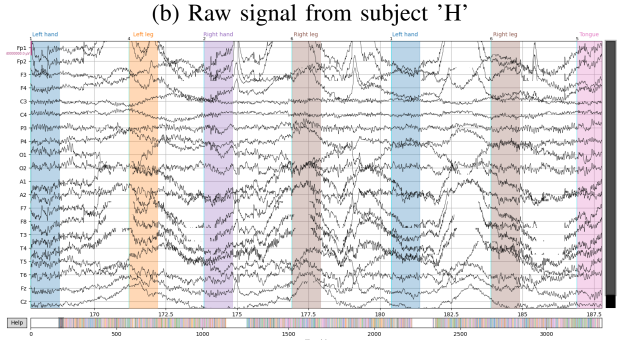
Indeed, after further inspecting the raw signals of the best accuracy and worst accuracy subjects plotted in Fig. 4, it is very noticeable that one signal is much cleaner than the other



(a) Raw signal from subject 'L'



(b) Raw signal from subject 'H'



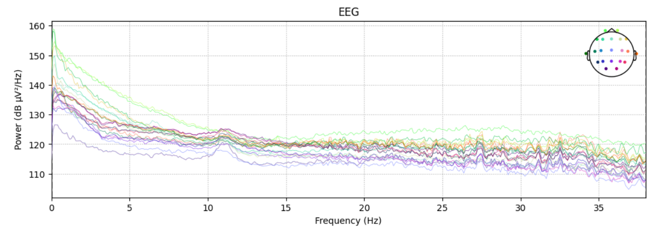
(c) Filtered signal from subject 'H'

Fig. 4: Comparison of raw and filtered signals from two subjects

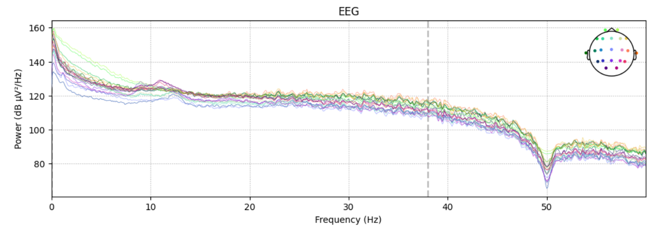
one. Furthermore, after applying the low-pass filter, the signal is smoother for Subject 'H', but it still seems to present a lot of noise,

We can also see by the Power Spectral Density (PSD) plots presented in Fig. 5 that subject 'H' has a higher power registered for the delta band (0.5 to 4 Hz) in most channels compared to subject 'L'. As such, experiments were made by applying to the data a Butterworth high-pass filter of 8-th order with a cutt-off frequency of 4 Hz to understand if these could improve the results of the model, but it had instead the opposite effect and actually resulted in poorer results, meaning that the frequencies in the delta band are helping the model to classify the action signals presented to the subjects, even though it is often characterized as artifacts due to this band being mostly associated to eye movements.

This might have also been given to the fact that the EEGNet is a network that is designed in such a way that makes it robust to noise and artifacts in the data [5].



(a) PSD from subject 'L'



(b) PSD from subject 'H'

Fig. 5: Comparison of raw and filtered signals from two subjects

Tab. 1 shows the architecture and number of parameters of the EEGNet applied to our problem.

The optimal choices of the hyperparameters obtained by the best accuracy when performing the cross-validation are represented in Tab. 2. The dropout rate was kept as suggested by the paper that purposed the EEGNet model [5] because we were using cross-validation.

In Tab. 3 we can see the best accuracies achieved when for the different strategies performed. It was already expected that the accuracies wouldn't be very high due to several factors, from the fact that we are trying to predict 6 classes to the important fact that we are creating a subject-independent model, which is trained on specific subjects and applied to entirely different subjects never seen by the model, with very different characteristics. We can see that the model with no data augmentation technique shows the least accuracy, while the model that has the best accuracy is the one with Bandstop width of 1.

VII. Concluding Remarks

A more extensive hyperparameter search can be computed, in order to both improve the model performance but to also further understand the impact of the different parameters in the robustness of the model. This can give further indications of how to prevent the model to overfit to the subjects used for training.

While we performed data augmentation using what could be considered more traditional approaches, the study in [6] has showed a significant increase in performance using generative networks to augment the data, in particular with GANs. This is a very appealing direction of improvement in the future.

Another interesting approach is using a Multilevel Weighted Feature Fusion architecture on the layers of the trained CNN, which achieved competitive results in subject-independency.

Layer (type)	Output Shape	Param #
Input layer	(None, 21, 210, 1)	0
TemporalConv2D	(None, 21, 210, 32)	2,400
BatchNormalization	(None, 21, 210, 32)	128
DepthwiseConv2D	(None, 1, 210, 512)	10,752
BatchNormalization	(None, 1, 210, 512)	2,048
Activation	(None, 1, 210, 512)	0
AveragePooling2D	(None, 1, 52, 512)	0
Dropout	(None, 1, 52, 512)	0
SeparableConv2D	(None, 1, 52, 64)	40,960
BatchNormalization	(None, 1, 52, 64)	256
Activation	(None, 1, 52, 64)	0
AveragePooling2D	(None, 1, 6, 64)	0
Dropout	(None, 1, 6, 64)	0
Flatten	(None, 384)	0
Dense	(None, 6)	2,310
Softmax activation	(None, 6)	0
Total params: 58,854 (229.90 KB)		
Trainable params: 57,638 (225.15 KB)		
Non-trainable params: 1,216 (4.75 KB)		

TABLE 1: Model architecture and parameters

Hyperparameter	Value
Kernel length	75
Temporal filters (F1)	32
Pointwise filters (F2)	64
Spatial filters (D)	16
Dropout rate	0.25

TABLE 2: Hyperparameters chosen for the EEGNet model

	Channel Dropout	Bandstop Width	CV Acc
1	N	1.00	0.5254
2	Y	0.00	0.5120
3	N	0.50	0.5069
4	Y	1.00	0.5013
5	Y	0.5	0.4977
6	N	0	0.4967

TABLE 3: CV accuracy metrics for model candidates (desc order).

Some difficulties encountered were related to the memory usage. This was solved after learning about the implementation of data augmentation strategies within the model architecture.

EEG data is indeed of very high complexity and there's a lot still to be explored and understood about it's behaviour. Although some strategies work in some datasets they may produce entirely unexpected results or even have neutral impact just by switching dataset. This makes EEG classification research a very challenging task and in need of extensive work to account for the significant variability in regards to session, recording method, environment conditions, but particularly to subjects, since the future demands that we produce methods of easier and more effective application for different individuals.

References

[1] M. Kaya, M. Binli, E. Ozbay, H. Yanar, and Y. Mishchenko, "A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces," *Scientific Data*, vol. 5, p. 180211, 10 2018.

[2] H. Altaheri, G. Muhammad, M. Alsulaiman, S. Amin, G. Altuwaijri, W. Abdul, M. Bencherif, and M. Faisal, "Deep learning techniques for classification of electroencephalogram (eeg) motor imagery (mi) signals: A review," *Neural Computing and Applications*, vol. 35, 08 2021.

[3] C. Rommel, J. Paillard, T. Moreau, and A. Gramfort, "Data augmentation for learning predictive models on EEG: a systematic comparison," *arXiv*, 2022.

[4] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for eeg decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, 2017.

[5] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brainâ€"computer interfaces," *Journal of Neural Engineering*, vol. 15, p. 056013, July 2018.

[6] F. Fahimi, S. Dosen, K. K. Ang, N. Mrachacz-Kersting, and C. Guan, "Generative adversarial networks-based data augmentation for brainâ€"computer interface," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 9, pp. 4039–4051, 2021.