# Single-trial stimuli classification from detected P300 for augmented Brain–Computer Interface: A deep learning approach

The main novelty of the proposed approach was to further distinguish the detected P300 components of ERPs based on the sensorial domain of their eliciting stimulus (i.e., visual vs. auditory) from a single trial.

Date Published: 5th of August 2022

## Dataset:

The produced dataset collected data referred to 22 healthy subjects, 11 men and 11 women, aged between 19 and 30 years. All of them proved to be right-handed, according to the proposed Oldfield Inventory test, wearing an elastic cap equipped with 126 electrodes, arranged according to the international standard.

Other datasets used in former researches can be found at [OpenBCI](https://openbci.com/community/publicly-available-eeg-datasets/)

## EEG Preprocessing:

* The preprocessing aims to discharge negligible EEG channels and improves signal to noise ratio
* EEG signals were acquired and analyzed via EEProbe recording software (ANT Neuro system, Enschede, The Netherlands). Stimuli presentation and triggering was performed using EEvoke Software for audiovisual presentation (ANT Neuro system, Enschede, The Nether\_lands). Digital amplifiers Synamps were used. The EEG was continuously recorded from 126 scalp sites at a sampling rate of 512 Hz. Horizontal and vertical eye movements were also recorded. Averaged ears served as the reference lead. The EEG and electro-oculogram (EOG) were amplified with a half-amplitude band pass of 0.016–70 Hz. Electrode impedance was kept below 5 kΩ, Signals coming from hEOG, vEOG, M1, and M2 electrodes were discarded in that not relevant for classification purposes. Baseline correction was applied to each EEG channel. This procedure consists of subtracting from each channel the average voltage recorded in the 200 ms preceding the stimulation. Since P300 is a low-frequency component, an offline band-pass filter was then applied, between 0.1 Hz and 20 Hz. In addition, artifacts rejection was performed, thresholding channels amplitude to ±50 μV
* 3D data representation outperforms the 2D one, accounting for both spatial and temporal EEG dependencies. Therefore, we encode data in a series of 21 × 21 matrices, n detail, each matrix represents the 2D projection of the electrodes cap. Accordingly, the pixel corresponding to an electrode collects the measured voltage in the instant the matrix is referred; pixels that do not correspond to an electrode are zero-padded.

## Classifier Architecture:

* A two-step hierarchical classification architecture was designed; a first classifier is trained to detect single repetitions of P300. Then, a second classifier further distinguishes the ERPs referred to the detected P300, as elicited by a visual or an auditory stimulus.
* We decide to combine the two architectures, producing a CNN–LSTM network. In detail, the produced hierarchical classifier comprises two instances of CNN–LSTM networks, each corresponding to a split node. This is the first time that the combination of these two networks’ structure has been used for P300 detection.
* Max pooling 2D and dropout layers are used to reduce data dimensionality, preventing overfitting
* Also, ReLU ,and batch normalization layers are leveraged to speed up the learning convergence and in reducing the sensitivity to initialization settings.
* The CNN portion structure is composed of an input layer, which passes input matrices to the convolutional layer, composed of 8 filters of size 3 × 3. Then, batch normalization and ReLU layers are applied. Each batch size is set to 64 samples. The activations produced by the following max pooling 2D layer are proposed as inputs to the LSTM structure, which learns the temporal data dependencies. This part of the network comprises a sequence input layer, which passes the instances to the following LSTM layer, composed of 512 units. Then a dropout of 0.5 is applied, and instances undergo another LSTM layer of 256 units. Finally, the classification is performed, leveraging a linear fully connected layer that weights the input coefficients and adds a bias vector; then, a non-linear softmax layer produces the actual predictions.
* Learning rate = 1𝑒 −4, optimizer = Adam

## Results:

* This Section first defines the metrics employed to evaluate the proposed BCI in terms of accuracy and interpretability. Then, the obtained results are presented and discussed. In detail, evaluation is performed considering two scenarios; in the first one, the system is trained and tested for each volunteer. Therefore, 22 hierarchical classifiers were trained and evaluated according to 10-fold cross-validation procedure.
* First Level that predicts target or not target:

Accuracy: Male = 80.7 ± 6.6 Female= 87.3 ± 4.3

F1-Score: Male = 78.1 ± 5.7 Female= 80.3 ± 4.7

Recall: Male: = 79.1 ± 6.2 Female = 85.8 ± 3.9

Precision: Male = 81.5 ± 11.9 Female= 92.15 ± 4.9

* Second Level that determines whether it’s a visual or audio

Accuracy: Male 92.8 ± 8.0 Female= 94.6 ± 4.5

F1-Score: Male = 90.9 ± 8.2 Female= 94.9 ± 6.0

Precision: Male: = 93.9 ± 8.1 Female = 92.3 ± 8.5

recall: Male = 92.3 ± 8.0 Female= 94.6 ± 4.5

# Fuzzy temporal convolutional neural networks in P300-based Brain–computer interface for smart home interaction

EEG patterns exhibit high variability across time and uncertainty due to noise. It is a significant problem to be addressed in P300-based Brain Computer Interface (BCI) for smart home interaction. It operates in a non-optimal natural environment where added noise is often present. In this work, we propose a sequential unification of temporal convolutional networks (TCNs) modified to EEG signals, LSTM cells, with a fuzzy neural block (FNB), we called EEG-TCFNet. Fuzzy components may enable a higher tolerance to noisy conditions. Our results reported a maximum classification accuracy of 98.6% and 74.3% using the proposed method of EEG-TCFNet in subject-dependent strategy and subject-independent strategy, respectively. Overall, FNB usage in all three CNN topologies outperformed those without FNB. In addition, we compared the addition of FNB to other state-of-the-art methods and obtained higher classification accuracies on account of the integration with FNB. The remarkable performance of the proposed model, EEG-TCFNet, and the general integration of fuzzy units to other classifiers would pave the way for enhanced P300-based BCIs for smart home interaction within natural settings.

Diagram

Description automatically generated

Date Published: 30th of December 2021

## Dataset:

The Speller test was performed on nine subjects, see further information in Fig. 2(a), divided into two groups: control and evaluation. The control group includes only healthy subjects from S01 to S06, while subjects S07 to S09 are patients with a poststroke condition. Subject S07 and S08 presented mild aphasia, especially S08 showed paresis in her upper limbs, and Subject S09 manifest severe apraxia. Data is collected from 16 channels.

## EEG Preprocessing:

* The signals are down sampled to 120 Hz to reduce computational cost. Then, a sixth-order Butterworth bandpass filter (1–15 Hz) is applied
* invalid spectral components were eliminated using Notch filters, and the elimination of outlier values followed the winsorization criteria

## Classifier Architecture:

* LeNet & LeNet + FNB

Table

Description automatically generated

* EEG-TCNet & EEG-TCNet + FNB

Table

Description automatically generated

* EEG-TCFNet (EEG-TCNet + LSTM + FNB) : proposed architecture

Diagram

Description automatically generated

Computation done on: Google Collaboratory

Programming language: Python

Optimizer: ADAM

Learning rate: 0.0001

Trained with leave-one-subject-out for subject independent

Trained with lave-one-session-out cross validation for subject-dependent

## Results:

Subject dependent resultsTable

Description automatically generated

Subject independent results

Table

Description automatically generated

ComparisonTable

Description automatically generated

Effect of downsampling on subject-dependent trials

Table

Description automatically generated