




Article

An Intrinsically Explainable Method to Decode P300 Waveforms from EEG Signal Plots Based on Convolutional Neural Networks

Brian Ezequiel Ail ¹, Rodrigo Ramele ^{1,*} , Juliana Gambini ^{2,3}  and Juan Miguel Santos ² ¹ Instituto Tecnológico de Buenos Aires (ITBA), Buenos Aires C1437, Argentina; bail@itba.edu.ar² Centro de Investigación en Informática Aplicada (CIDIA), Universidad Nacional de Hurlingham (UNAHUR), Hurlingham B1688, Argentina; juliana.gambini@unahur.edu.ar (J.G.); juan.santos@unahur.edu.ar (J.M.S.)³ CPSI—Universidad Tecnológica Nacional, FRBA, Buenos Aires C1041, Argentina

* Correspondence: rramele@itba.edu.ar

Abstract: This work proposes an intrinsically explainable, straightforward method to decode P300 waveforms from electroencephalography (EEG) signals, overcoming the black box nature of deep learning techniques. The proposed method allows convolutional neural networks to decode information from images, an area where they have achieved astonishing performance. By plotting the EEG signal as an image, it can be both visually interpreted by physicians and technicians and detected by the network, offering a straightforward way of explaining the decision. The identification of this pattern is used to implement a P300-based speller device, which can serve as an alternative communication channel for persons affected by amyotrophic lateral sclerosis (ALS). This method is validated by identifying this signal by performing a brain–computer interface simulation on a public dataset from ALS patients. Letter identification rates from the speller on the dataset show that this method can identify the P300 signature on the set of 8 patients. The proposed approach achieves similar performance to other state-of-the-art proposals while providing clinically relevant explainability (XAI).

Keywords: XAI; BCI; EEG; P300; ALS; waveform; deep learning; CNN

Citation: Ail, B.E.; Ramele, R.; Gambini, J.; Santos, J.M. An Intrinsically Explainable Method to Decode P300 Waveforms from EEG Signal Plots Based on Convolutional Neural Networks. *Brain Sci.* **2024**, *14*, 836. <https://doi.org/10.3390/brainsci14080836>

Academic Editor: Andrej M. Savić

Received: 19 July 2024

Revised: 15 August 2024

Accepted: 16 August 2024

Published: 20 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The area of brain–computer interfaces (BCIs) involves the quest to create an alternative communication channel between the central nervous system and a machine not based on natural pathways [1]. Recent advances have shown impressive achievements, like controlling a prosthetic limb with the mind [2,3] or a monkey playing the computer game Pong [4]. Although these outstanding developments are based on invasive electrodes, a steady improvement albeit with modest results is also being attained with noninvasive approaches, particularly through the use of electroencephalographic (EEG) signals [5].

How to harness the information from these signals? The P300 is an event-related potential (ERP), a positive deflection that appears on EEG signals around 300 ms after the onset of an unexpected stimulus. It can be cleverly used to implement a speller device that exploits the appearance of this waveform indicating the event which a person is paying attention to. BCI systems based on P300 need to identify this voltage variation from a noisy EEG signal, and by doing that they can decode the message a person is trying to convey [6,7].

In terms of potential medical applications, patients with severe cases of amyotrophic lateral sclerosis (ALS) [8] can present a locked-in state, in which they are unable to move any of their muscles to communicate. Thus, traditional alternative augmentation communication (AAC) devices [9], that harness information from any remaining muscle activity, are not effective. Creating a BCI device may be the only remaining choice to allow the person to connect with the outside world and help them to improve their life quality [10,11].

Particularly in high-stakes systems, like those for healthcare applications, any inference model based on automatic decision-making requires providing an explanation or interpretation about how the system decoded the information [12]. For ALS patients, the issue of understanding the stability of the P300 signal is paramount and goes beyond the detection of this pattern, having also clinical endpoints like aiding the understanding of neurodegenerative disease [13,14].

A plethora of methods have been applied to decode EEG signals [7,15] and particularly to decode P300 components [16]. However, only a few have focused on the idea of determining the presence of the P300 ERP response based on the waveform that can be obtained from a signal plot [17,18].

Concurrently, the success of deep learning in several fields [19] has opened the possibility of using this same technique to identify the complex patterns that can be found in EEG signals [20–25]. Particularly, convolutional neural networks are successful techniques that have pushed this field ahead [26] and are often used in computer vision or speech recognition applications. Moreover, they have also been applied extensively to decode EEG signals [27–32]. Two approaches are important to note. The first is called BCINet, which is a convolutional neural network used to extract workload markers from EEG signals [33]. The other approach, called EEGNet [34], is also a convolutional neural network designed generically to process EEG data, using the convolution operation as a way to implement multiscale filter-banks [35].

Despite these advancements, the utility of DL in processing EEG signals, and healthcare information in general, is still debatable [36–39]. This is mostly due to two reasons: (1) The training steps of DL models require as many data samples as possible to achieve convergence and to generalize better, though in healthcare applications, data from patients are not readily available, and are much harder to obtain. (2) The features learned by using deep learning models face the problem that they cannot be easily understood [40], a characteristic which is often called a black box problem. Computational inference systems can have the property of being interpretable or explainable (XAI). Explainability refers to the capacity of a system to reveal the sequence of decision-making steps that the model followed throughout the inference process, while interpretability refers to the feasibility of rendering the prediction understandable to humans [41,42]. Computational inference systems that are not readily interpreted or explained are often called black box systems [43].

The work presented here circumvents this problem because the DL-CNN is used for what it is known to already perform well: decoding information from images. This decoding is based precisely on something intrinsically interpretable that we humans also perform well, detecting patterns by the visual identification of waveforms.

This proposal expands on the work performed for the dissertation [44] and follows the line of methods proposed in [18,45] and in the dissertation [46], where the waveform is also analyzed but using traditional tools from computer vision. Hence, the aim of this work is to analyze P300 signals by an intrinsically interpretable procedure which automatically detects this component from the signal plot, something that a clinician or physician can easily visually corroborate [47]. To do so, this work proposes to use deep learning (DL) techniques, particularly convolutional neural networks (CNN), which have been proven very efficient in extracting information from images, in this case images of signal plots. At the same time, this proposal tackles one of the most important issues of applying DL techniques to health information, which is the black box nature of neural network architectures [43]. It emphasizes the usage of a new form of intermediate feature engineering in terms of producing actionable analytics [48], insight data that can be extracted from the problem, and that have a certain meaning in terms of clinical knowledge. This is implemented by a pragmatic approach tailored for this particular problem where data are scarce [49].

This work proceeds as follows: Section 2 explains the details of the public dataset of ALS patients, and describes the architecture designed to process it. In Section 3, the results are expounded and compared against other similar procedures. Section 4 presents a discussion. Finally, in Section 5, conclusions, limitations, contributions and future work are highlighted.

2. Materials and Methods

2.1. P300 Experiment

The dataset used in this work is the 008-2014 published under the BNCI-Horizon 2020 website [50]. The original experiment was performed by Riccio et al., 2013, where a group of eight individuals with confirmed ALS disease were tasked to spell 7 five-letter words (35 total letters) with a standard P300 matrix, which can be seen in Figure 1 (experiment details and ethical approval information available in [51]). This P300-based speller device works by flashing alternately the rows and columns of the matrix. The flashing, or letter intensification, serves as a visual stimulus that triggers the P300 response on an EEG signal segment, and by detecting it, the selected letter can be inferred from the intersection of the row and column.

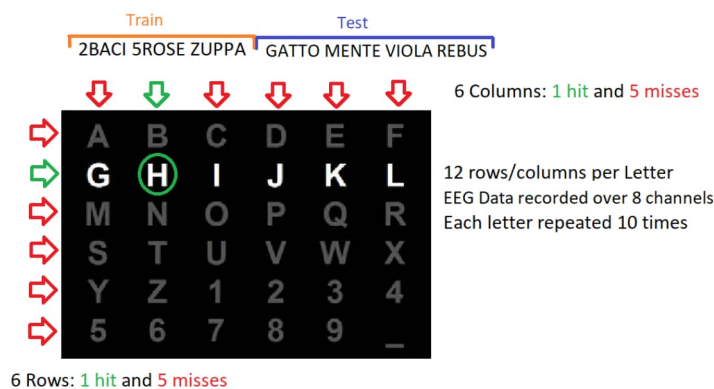


Figure 1. P300-speller matrix used in the experiment. The 7 five-letter words, divided into train and testing, are shown on top. These are used in the P300 experiment for the copy-spelling task.

Each letter spell attempt is composed of 10 flashes of the 6 rows and 10 flashes of the 6 columns of the P300 matrix. Each flash lasts for a total of 0.125 s, followed by an inter-flashing pause of the same duration. After every 120 flashes, an inter-trial pause is performed before moving to the next attempt [6].

This experiment was performed with an 8-channel EEG device (g.Mobilab, g.Tec, Austria), with a sampling frequency of 256 Hz with electrodes placed at the Fz, Cz, Pz, Oz, P3, P4, PO7, and PO8 channels according to the 10/20 international system. The reference for all the channels was the right earlobe and they were grounded to the left mastoid. The software used for implementing the speller and the processing was the BCI2000 [52]. The eight subjects were instructed to complete a copy-spelling task, which means that they had to spell a predetermined set of words that was instructed beforehand. In the original experiment, and in the approach used in this work, the objective is to use the first three words for calibration/training, while the remaining four are used for letter identification.

2.2. BCI Simulation

The task of decoding information from brain signals inherits practices from machine learning (ML). Nested cross-validation [53] is used in ML to reduce overfitting bias and to increase the independence from the dataset that is used as calibration. However, the brain data used in BCI are extracted from a person who is performing a task and whose signals are changing while trying to adapt to this operation. Hence, taking for granted exchangeability [54], mixing the dataset, shuffling the sessions and trials is at least a challenging assumption. BCI simulation, on the other hand, is not very well defined in BCI research, but its practice, without naming it, has been the regular approach for BCI competitions. It consists of reproducing the operational sequence that was utilized to generate the dataset. Therefore, the experiment is replicated offline using the training information to train or calibrate a classifier, and then to classify the testing signals as if they were generated at that same moment. In order to simulate a real use-case of a subject

actually using the interface, the training is performed with the first letters and then used to decode the remaining letters simulating the online procedure.

2.3. Signal Preprocessing and Plot Generation

The original dataset was sampled at 256 Hz. We kept the same sampling frequency (i.e., without any spectral filtering). The next step involves processing the signal to enhance the signal-to-noise ratio of the P300 component. Finally, the processed signal is plotted to generate an image, which will contain the visually relevant P300 waveform.

Electroencephalographic signals are complex, non-stationary, noisy, and often contaminated by artifacts, originating either as biological sources from the human subject, from the environment, or from the recording device itself [55]. In order to detect the P300 component, it is important to maintain a high signal-to-noise ratio (SNR), which is defined as the information content between the P300 element versus all the rest. When the SNR is low, the target signal is harder to detect, since it is obfuscated by noise.

A common approach to improve the SNR is by using *signal averaging*. This technique can be applied for time-locked signals where the timing of the signal is known, the noise and signal are not correlated, the noise has a Gaussian distribution with zero-mean, and the signal is consistent if the experiment is performed multiple times [56].

A raw single-channel signal can be characterized as a time point sequence. This sequence is obtained by a digitalization process, at a certain sampling frequency F_s which depends on the electrophysiological device [57]. The original EEG stream can be divided into segments or epochs $x_i(t), 1 \leq t \leq L; L$ is a fixed length and rescaled to have zero-mean, which we assume meets the conditions mentioned earlier. These segments can be described as:

$$x_i(t) = n_i(t) + s_i(t), 1 \leq t \leq L, i = 1, \dots, N \quad (1)$$

where x_i is the signal segment on the i th repetition of the trial, the flashing of a row or column, and L is the length of the segment. This signal segment is composed by a noise term n_i and a timed locked signal term s_i . If we repeat the same experiment N times and average them, the resulting averaged segment is:

$$X(t) = \frac{1}{N} \sum_i^N (n_i(t) + s_i(t)). \quad (2)$$

Since the noise can be considered random with a zero-mean, and the time locked signal has a similar pattern throughout the segments, we can approximate:

$$X(t) \approx S(t), \quad (3)$$

where $S(t), 1 \leq t \leq L$ is the average of the time-locked signal segments. In this case, we have removed the noise and have a clearer signal. It should be noted that this equation can only be considered valid once N is large enough since we need an infinite number of repetitions to truly eliminate the noise.

One remarkable point is that when averaging independent signals and $Var(x_i) = \sigma^2, \forall i = 1, \dots, N$, with $N > 1$, the variance of the averaged signal is lower than the variance of each individual $x_i, i = 1, \dots, N$:

$$Var\left(\frac{\sum_{i=1}^N x_i}{N}\right) = \frac{1}{N^2} \sum_{i=1}^N Var(x_i) = \frac{N\sigma^2}{N^2} = \frac{\sigma^2}{N} < \sigma^2. \quad (4)$$

This aspect is relevant in this case because the shape of the waveform depends on the variance of the signal and the P300 experiment naturally produces a skewed distribution of segments between the classes. Hence, any classifier based on the variance, trained on this problem, could discriminate between images produced with a different number of samples, instead of detecting the difference based on the shape of the P300 component. To

avoid confounding these two things, the averaged signal $X(t), 1 \leq t \leq L$ is normalized by applying the z-score transform [58], effectively normalizing the variance on all the signal segments from both classes.

Plotting an averaged signal segment entails a digitalization process and it produces a binary image per $X(t), 1 \leq t \leq L$, with the trace representing the time-varying signal [59]. This binary image \mathcal{I} can be constructed based on Equation 5,

$$\mathcal{I}(z_1, z_2) = \begin{cases} 255 & \text{if } (z_1, z_2) = (\gamma_w t, \lfloor \gamma_h X(t) \rfloor + Z) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $1 \leq t \leq L$, $Z = \frac{H_y}{2}$, with $H_y = 2 \gamma_h \left| \max X(t) - \min X(t) \right|$ and $W_x = \gamma_w L$ being the image height and width, respectively. The symbol $\lfloor \cdot \rfloor$ is the integer part operator. The coordinates z_1 and z_2 are on the horizontal and vertical axes of the image, respectively; z_2 increases from top to bottom, so the $(0,0)$ position is on the upper-left corner of the image. The amplitude scale factor γ_h and time scale factor γ_w are used to determine the image size and, at the same time, the image resolution. In order to complete the trace of the signal plot, the isolated points produced by Equation 5 are connected using the Bresenham [60] algorithm, which performs a linear discrete interpolation between the pixels. This scheme produces a black-and-white plot of the signal with 255 being white and 0 black. There is one image per channel per segment.

In this work, the averaged signal segments $\{X(t), 1 \leq t \leq L\}$ have a length of 800 ms sampled with a frequency $F_s = 256$ Hz, resulting in a total of $L = 204$ datapoints.

This generates an image where the plot fills the whole width, and leaves 25% of air on the top and bottom. The parameters γ_w and γ_h are scaling values that were set at 2 and 30, respectively, and Z is the vertical center of the image and corresponds to the position on the image where the signal is zero. As the max and min of each signal are different, the height of the generated images varies from image to image.

Finally, an image rescale is implemented using bilinear interpolation to 150×150 pixels, generating squared images. This is due to the fact that we configured the proposed neural networks to use inputs with that same dimension, and as mentioned earlier, the height of each image depends on the maximum and the minimum value of the signal. This rescaling provides a standard size for all of the signal plots.

These images are used following two different pipelines, as can be seen in Figure 2. The first alternative (top of Figure 2) includes a channel selection procedure based on picking the best performing channel to plot the input image, while the second alternative (bottom of Figure 2) uses a different approach where the information from all the channels is used together at the same time to create a multichannel waveform image. This image is used as input to a CNN that will be trained to classify images containing the waveform versus those that do not.

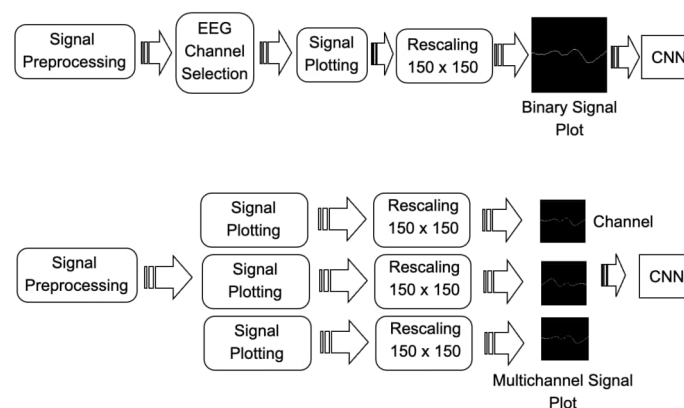


Figure 2. Two alternatives pipelines are used: (**top**) with channel selection, (**bottom**) bundling the information from all the channels together.

2.4. Neural Network Architectures

This work proposes three neural network architectures to identify the pattern of the P300 component on the signal plot. They are based on VGG16, one of the first deep convolutional neural network architectures, which was used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVR) competition in 2014 [61]. It is characterized by several layers, but at the same time, it keeps a simple structure based on the convolution operation. Convolutional neural networks are bioinspired deep learning neural networks, organized in a sequence of layers, that process an input to provide a desired output. The transition of the information from one layer to the next is mediated by the convolution operator, where the values of the convolutional kernels are the weights that are being optimized to achieve the desired output. This acts like a filter that needs to be learned. The sequence of convolutional layers is intermixed with selective decimation operations called pooling [62]. This combination of operations is particularly well suited to process visual images, in our case, visual plots of signal waveforms.

The image plot generated by the previously described procedure is used as input to each CNN for the first two proposed architectures. For the third one, the eight plots, one per channel for one segment, are bundled together in a single input.

Regarding the training, it is performed on each architecture with the first 15 letters of the speller (i.e., the first 3, 5-letter words), and the network is used to predict the remaining 20 letters (4, 5-letter words). The performance of the network is calculated by finding the row and column with the highest chance of containing the P300 signal according to the prediction of the neural network. This can be used to determine the predicted letter, which is then compared to the original expected letter from the experiment. These predictions are performed by training using the images derived from each channel separately, and then finding the channel that performs the best for the first two architectures, or combining all the images together for the third one.

2.4.1. VGG16

The first version of the neural network (Figure 3a) is based on vanilla VGG16. It uses a 3×3 filter with a stride of 1 and uses padding to keep the same spatial dimensions, followed by a MaxPool layer of 2×2 size and stride of 2, and follows this arrangement of convolution and MaxPool layers all the way throughout the whole architecture, for the convolutional and MaxPool sets. The original VGG16 has 2 fully connected layers, while for this particular case, we are using 4. In the end, the last layer with only one neuron is activated with a sigmoid function to perform the binary classification. The network implementation has 6 convolutional layers with depths of 1, 32, 64, 128, 128 and 256. The stride in the MaxPool layer reduces the spatial size in each iteration, leaving a spatial size of 150, 75, 38, 19, 10, 5. After the convolutional layers, there is a dropout layer with a drop rate of 0.5, followed by a flattened layer to prepare the data for the dense layers. The 4 dense layers have a size of 6400, 1024, 512, 256, and 1 unit, with the last one having a sigmoid activation. The optimizer used for the training procedure is Adam [63], and the loss function is the mean square error (MSE). The learning rate used is adjusted to 5×10^{-4} , which deviates from the recommended 3×10^{-3} [61]. Finally, the batch size is adjusted to 20. The number of trainable parameters for this model is 7, 209 or 824.

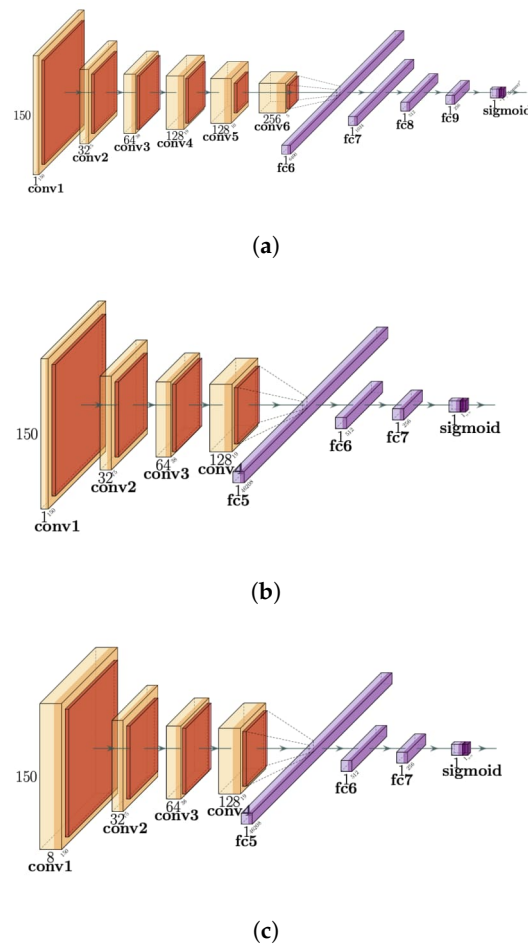


Figure 3. The three architectures proposed in this work. (a) VGG16: First version of the CNN. The input is a binary image plot of a signal of size 150×150 . Then follows a set of 6 convolutional layers, followed by 4 fully connected layers, and a final layer activated using a sigmoid function. (b) The second, SV16 is similar to VGG16, with the same input but has 4 convolutional layers and 3 dense layers. (c) Finally, MSV16 has the same architecture as the SV16, but now the input layer is modified for an 8-channel input, with one binary image plotted with a signal waveform per channel. MaxPool layers are shown in dark orange.

2.4.2. SV16

The second approach is the Small VGG16 (SV16), shown in Figure 3b. The first change is to reduce the network size, so 2 convolutional layers and 2 fully connected layers were removed. Leaving the convolutional layer depths at 1, 32, 64, and 128, with spatial sides of 150, 75, 38, and 19. The flatten and dropout layers are kept, and then, finally, 3 fully connected layers of sizes 46, 208, 512, and 256 are added. Another addition in this version is an early stopping at 7 epochs. The remaining components of the architecture are kept unmodified, i.e., the padding, optimizer, batch size, filter size, and stride. The increase in fully connected layers brings an increase in terms of the number of trainable parameters to up to 23, 790, and 048.

2.4.3. MSV16

Digital images are encoded by pixels in a square grid with a certain height and width. Grey images use one value per pixel, and this value determines the grey intensity on the screen. Color images on the other hand, use more than one value per pixel. For instance, these values can be representations of red, green and blue intensities. Furthermore, this idea could be extended to even more color channels, each representing an independent intensity, just like the colors. Particularly, convolutional neural networks are designed to analyze

color images, where each color represents an independent channel of information. With the Multichannel Small VGG16 (MSV16), represented in Figure 3c, instead of selecting just one EEG channel, and training the network with the plots produced from this EEG channel, they are bundled into a single multichannel waveform image, which is the combination of the 8 plots, each one per channel acting as if it were a different color. Then the network is trained with this color image plot as input. This requires the modification of the input layer to have a depth of 8 (instead of 1), but the subsequent layers are left intact, preserving the same number of trainable parameters, 23,790,048.

Nonetheless, changes are required to the network architecture to cope with this change in the input data: the batch size is reduced to 6, as having a greater batch size imposes a higher memory requirement and the learning rate is readjusted to the recommended value for VGG16 of 3×10^{-3} .

2.5. Dataset Balancing

The cognitive oddball paradigm used to trigger the P300 signal inherently has the problem that the produced dataset is unbalanced [64]. In this particular case, for each letter, there are 20 segments that belong to the hit class and 100 that do not. This skews and biases the training of the CNN. Hence, we implemented a quick and effective solution of resampling the minority class on training, which is to pick randomly the same number of samples for each class to fill the network batch size and discard the rest.

2.6. Software and Hardware

The code for all the experiments runs on an HP Pavillion laptop with an Intel I7 @2.8GHz processor. The available RAM is 16Gb, and the entire stack runs on CPU. No GPU was required.

The software for the signal segmentation and processing is written in Python, using a modified version of the repository EEGWave, which is publicly available on the CodeOcean platform [18]. It uses the MNE library [65] for segmenting the data and some operations with numpy [66] for signal processing procedures. On the other hand, the generation of the plot of the signals is written in C++ using OpenCV, and all the neural network architectures are created using the very efficient Tensorflow's API for C++, based on Benny Friedman's implementation and article [67]. The full code is public and can be found at <https://github.com/shipupi/BciSift/> (accessed on 19 August 2024). For the sake of replicability [68], all the software and data for this experiment are fully available online.

3. Results

In this section, the results of applying different alternatives to decode the speller letters are shown. By using plain letter identification rates, meaning the percentage of correct spelled letters over the 20 available number of letters to spell, we obtain a solid metric that unequivocally reflects the capacity of each method in identifying the spelled letter directly from the brainwaves.

To compare the results of applying the proposed methods with another alternative that does not use artificial neural networks, we utilize the algorithm presented in [45], based on the scale invariant feature transform (SIFT). This method can find distinctive image features using a histogram of oriented gradients from pixel intensities. In addition, it has the advantage of being invariant under scale and affine transformations [69].

Figure 4 shows the results of computing the letter identification rate, by measuring success in the letter prediction. The x-axis represents the increasing number of letter intensifications that are used to calculate the averaged ERP response (i.e., using more information). Figure 4a–c represent the results obtained by using VGG16, SV16 and MV16, respectively. For the sake of comparison, results obtained from the SIFT method [45] are shown in Figure 4d.

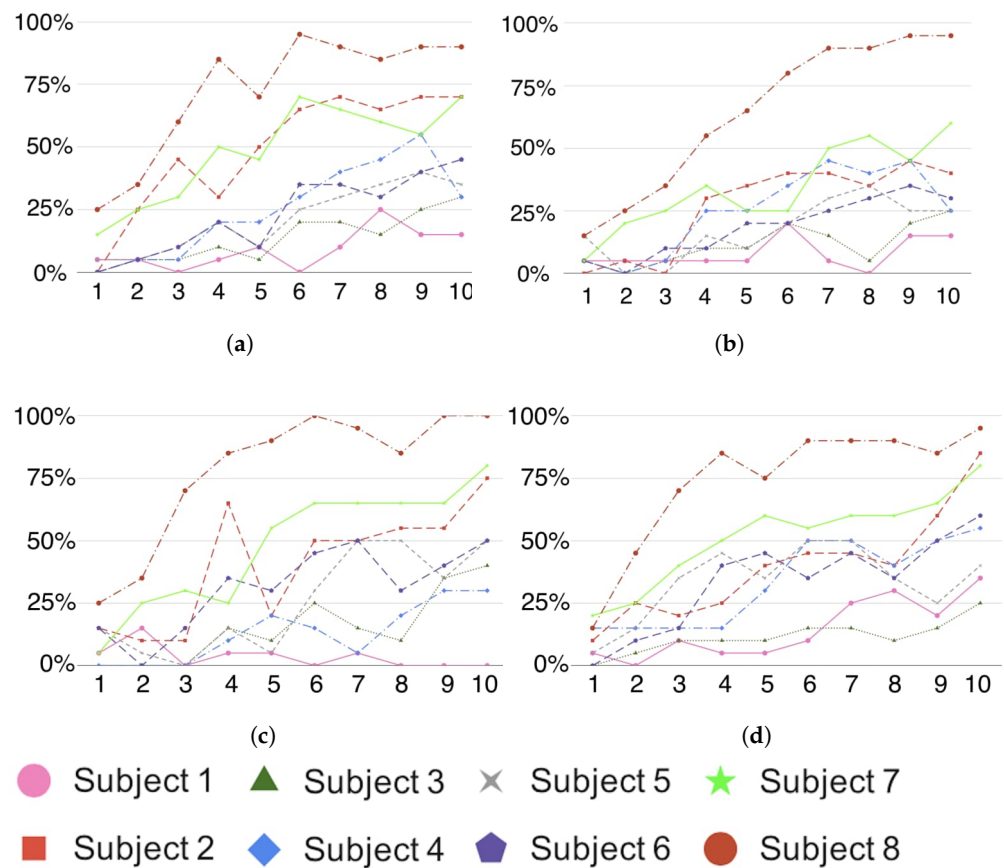


Figure 4. Side-by-side comparison of letter identification rate (y) per number of letter intensifications (x) for all the three architectures proposed in this work and one more for comparison: (a) VGG16, (b) SV16 and (c) MSV16, (d) SIFT method [45].

It can be observed that the three networks were successfully trained with the generated plots. The learning curve for subject 5 of the dataset at an intensification level of 4 can be seen in Figure 5. Figure 5a,b show the loss value, training and validation accuracy of using VGG16 and SV16, respectively. Comparing both, the accuracy value for the validation set is much higher for the SV16, while the accuracy value for the training set remains at or close to 100%.

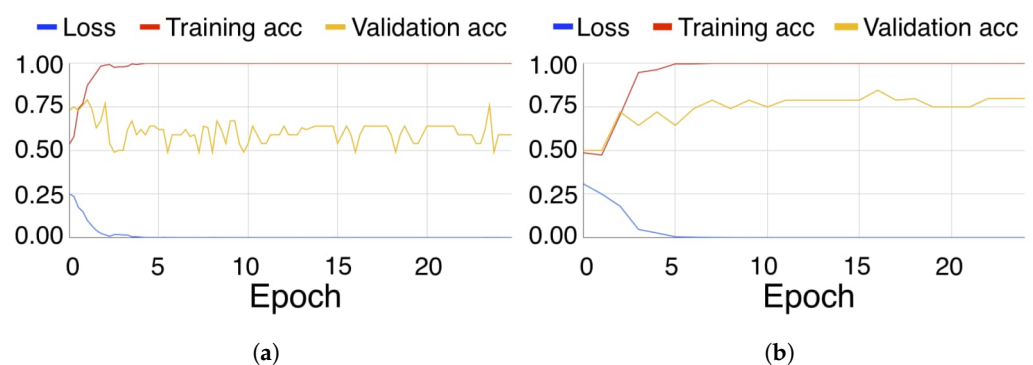


Figure 5. Learning curve, showing loss, training accuracy and validation accuracy of subject 5 on channel PO8 with an intensification level of 4 using (a) VGG16 and (b) SV16.

Letter identification success rates for 10 letter intensifications are shown in Table 1 for each subject from the experiment dataset. Stepwise linear discriminant analysis (SWLDA) values are based on the algorithm reported and used by Riccio et al., 2013 [51]; SIFT are

the values obtained by using SIFT descriptors. The table includes results using the vanilla SVM method [45], BCINET [33] and EEGNET [34].

Table 1. Letter identification rates in percentages for SWLDA, SVM, EEGNET, BCINET, SIFT, VGG16, SV16 and MSV16 for the public dataset [51].

Subject	SWLDA	SVM	EEGNET	BCINET	SIFT	VGG16	SV16	MSV16
1	45	40	50	45	35	15	10	0
2	30	50	30	55	85	70	50	75
3	65	55	70	65	25	30	30	40
4	40	50	60	55	55	30	40	30
5	35	45	40	45	40	35	50	50
6	35	70	60	75	60	45	40	50
7	60	35	90	80	80	70	65	80
8	90	95	100	95	95	90	95	100

4. Discussion

Overall, the obtained results show that all three proposed architectures achieve a success rate while identifying spelled letters, which is comparable to other similar published works on this same dataset. Moreover, success rate values are consistently increasing along the intensification level. We did not find any significant difference between all the eight methods listed in Table 1 (Quade Test, $F = 1.76$, $p = 0.11$) [70].

Nonetheless, there are important points to highlight. First, the advantage of using an already established model like VGG16 is that it reduces the number of hyperparameters that need to be tinkered with, as all the filters and strides are known to be effective for processing images. Finding a suitable architecture and its parameters is a daunting task. It is clear from the literature that there is no certainty as to which hyperparameters to use when dealing with information from EEG signals [71].

Another aspect to note is that the three proposed architectures required some data balancing mechanism in place, otherwise the neural network was unsuccessful in converging (i.e., not reducing successfully the training error). This is also found in similar works, either by subsampling the majority class or by resampling the minority class on training [71].

Regarding the VGG16 architecture, the main issue we have found is overfitting. The training sets are being learned perfectly, reaching a 100% accuracy, but the accuracy on the validation set does not increase and oscillates around 60%. The SV16 version attempted to address this issue by adding early stopping and reducing the number of layers, as shown in Figure 3b.

The biggest increase in performance in this work is the MSV16 version, with the inclusion of all the plots of the EEG channels into a single multichannel waveform image plot. This third and final version of the network included both the improvements made in SV16, as well as the addition of multi-channel classification, and a smaller batch size. Additionally, this approach can help to understand spatial asymmetries in the signal, which can have clinical implications. This last version showed an accuracy increase in six out of eight subjects over the other two versions. It surpasses SIFT in three out of eight subjects and performs equally in one. It does, however, seem to underperform on subjects 1 and 4 compared to earlier versions and SIFT. But it also reaches a 100% accuracy on subject 8, as was the case for EEGNET. Also, by looking at the accuracy per intensification level, subject 8 achieved over 90% accuracy in only five intensifications, meaning that a robust speller could be established with a much faster transmission rate. This can also be visually seen in Figure 6 where a similar shape from images (a) and (c) is what the neural network is actually identifying, what it is actually seeing, the shape of the P300 waveform. The best improvement was that it allowed the combination of all the different EEG channels to work together, achieving a higher performance than each channel separately [37]. This tendency could potentially be greater for EEG signals recorded with more than eight channels.

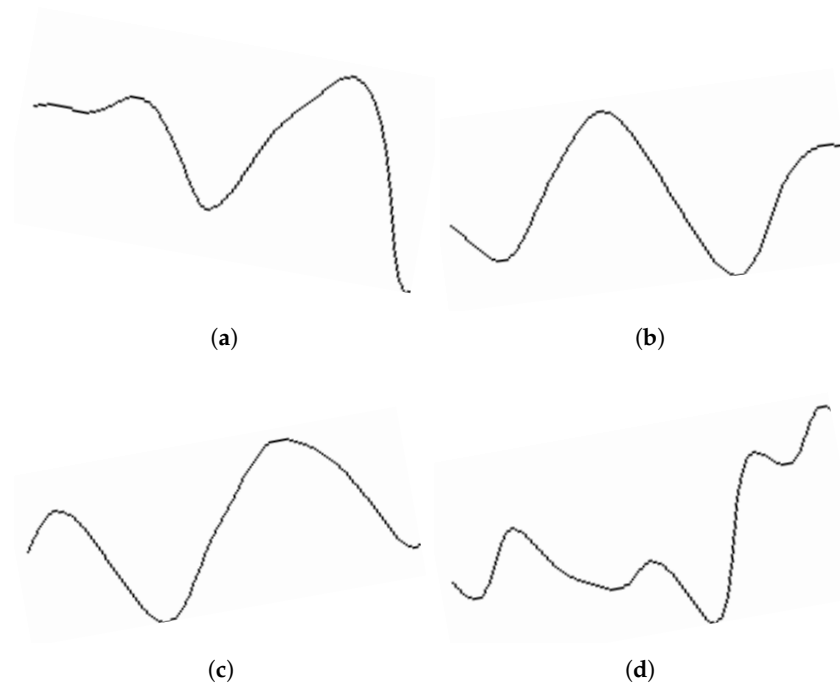


Figure 6. Images of P300 waveforms (a,c) from the first 0.700 ms of stimulus-locked segments of channel Cz of subject 8, obtained by averaging the signal segments triggered from 5 intensifications. Images (b,d) are the obtained shape when this waveform is not present.

The results of the three approaches show clearly the importance of the signal averaging procedure in increasing the signal-to-noise ratio (SNR). On the other hand, the downside of signal averaging is that repetitions of the experiment take more time, thus slowing down the information transfer rate (ITR). As BCI devices are communication systems, ITR is an important measure of performance [72], with units in bits-per-second (bps) and a method to measure the advancement of this technology. Out-of-the-box BCI systems could obtain very low transfer rates, of around only 5 bits per minute or 0.08 bps [73]. For instance, the system proposed in this work, using the MSV16 version, could achieve an ITR of 0.6 bps, at 5 intensifications per letter. More sophisticated systems, tailored for a specific group of patients using brainwaves in non-invasive SSVEP Speller can achieve 13 bps [74]. BCI systems have a wide inter-subject variability, so there are works reporting faster rates but their generalization must be carefully considered [75]; thus, it may be difficult to reach the 13 bps value for every person and even for the same person in different settings and across sessions. Invasive systems with implanted electrodes are faster, reaching from 5 up to 15 bps or even more [4]. Some specific use-cases require a very high bandwidth (>15 bps), for instance, the one needed to control all the degrees-of-freedom (DOF) of an arm prosthesis [2,76,77]. For reference, keyboard typing is 16–20 bps (200 words per minute), while speaking is 39 bps [78].

Finally, control experiments were performed to ensure the validity of the obtained results. First, while performing the signal averaging procedure, the standard deviation of the averaged signal segment $X(t)$ is calculated, and it is asserted that it is indeed lower than the one for each signal segment $x_i(t)$. Additionally, as long as more segments are used, it is verified that it is progressively reduced. To corroborate that the obtained results are not due to chance, we ran the experiment on all the subjects but with all the labels randomized on the training dataset. The expected outcome of this procedure is that the predictions for the letters are reduced to the chance level of $1/36 \approx 3\%$. This is a procedure inspired by the Boruta feature selection [79]. We can see this in Figure 7, where the accuracy of the prediction of the MSV16 network with random labels on subject 8 is oscillating around the chance level threshold, while the accuracy without randomizing the training labels

goes back to the reported value. This shows that there is indeed a generalization being performed by the neural network.

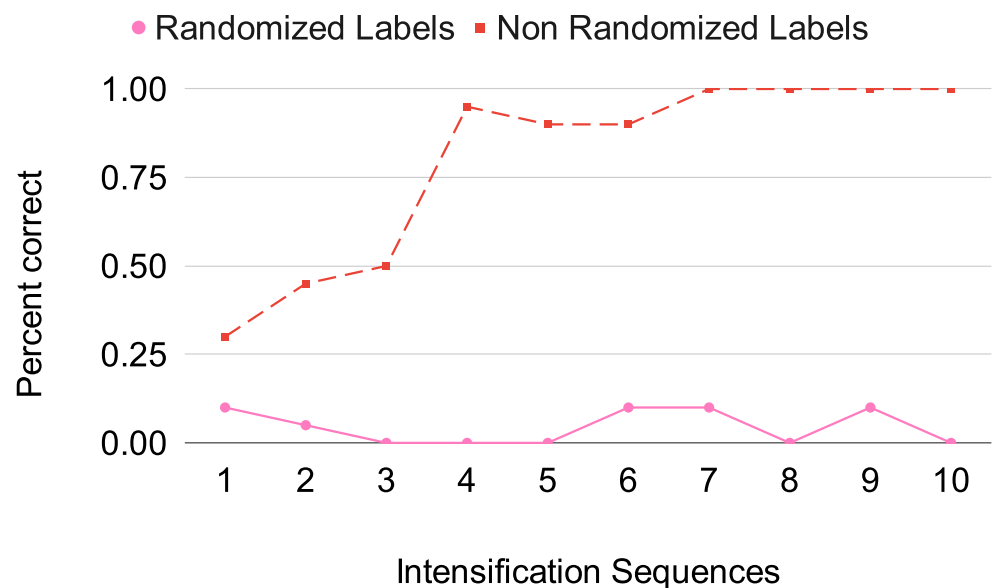


Figure 7. Accuracy of MSV16 on subject 8; the red dotted line shows a normal prediction using real labels on the training set, while the pink line shows the predictions of the CNN by training it with randomized labels on the training set. We can see that the accuracy on the pink line hovers around the chance level of 3%.

Extensive work has been conducted about XAI to develop methods that aim to alleviate the black box nature of DL models [80]. CNN feature maps [81,82] are a popular approach that enables some form of explainability by analyzing decision regions in the hidden layers of the network and inferring what their contribution is to the final decision. However, they create an even wider gap that complicates the understanding and communication from the human perspective [83], and may hamper even further trust in the system [84].

Moreover, there is an existing correlation between the stability of the P300 signal pattern and indexes associated with ALS disease prognosis [13,14,46]. For instance, the images of P300 waveforms shown in Figure 6a,c correspond to a patient where it was verified that the obtained P300 signal is stable [46]. This is also what the CNN network detected through higher letter identification rates. Additionally, quantitative parameters, like the amplitude or latency of the ERP, can be directly extracted from those images themselves, providing a straightforward way to perform the automatic analysis and the qualitative visual analysis of the waveforms. Furthermore, in a previous work [46], where the waveform was analyzed by the SIFT method, we performed exactly this same experiment with a control group of healthy individuals and we found no evidence of significant changes in terms of the achieved performance on the experiment between the group of healthy volunteers and this group of ALS patients.

We believe that the approach proposed here could eventually be a feasible tool for clinical settings that aim to detect ERP abnormalities in an automated way, connecting the reasons for the performance of this deep learning detection procedure with the shape of the ERP waveform. In particular, there is evidence in the literature that ERP abnormalities in the waveforms can be detected in ALS patients in early stages of the disease without major cognitive deficits [14].

5. Conclusions

The purpose of this work is to provide an intrinsically explainable method for convolutional neural networks to identify the P300 signal pattern. This can aid in understanding clinical insights that are particularly relevant to ALS patient populations. It is important to

remark that the proposed method provides a straightforward way for a neural network to provide explainability. By trying to process the information in a similar way that humans deal with this information, actionable analytics can be extracted and are ready to be used in the problem at hand. We believe it is highly desirable to offer a human-centered approach to explainability, emphasizing insights that could be understood by professionals and clinicians and not the other way around [85]. We force the automation to talk in human terms.

Additionally, we have shown that the obtained performance is at the same level as other methods that lack these XAI characteristics. Moreover, this approach is very general, because it can be used to decode any signal that can be represented in a plot, and the discriminating power can be encoded by the waveform inside the plot.

One of the reasons why deep learning techniques are highly successful is because they can produce tools upon which other solutions can rest. Using the original structure of VGG16 as a baseline foundation optimized the effort involved in the time-consuming and iterative process of optimizing hyper-parameters and network architectures [86].

One aspect to specifically remark on is that this procedure combines information from different channels at the same time, which perfectly matches what may be appropriate for CNN. We have verified that this approach generates better results than those obtained with signal plots images that only reflect the information from just one channel. And even though the data used are scarce for each subject, the CNN was able to obtain enough samples to be trained to effectively discriminate the P300 ERP.

Several limitations and considerations need to be taken into account. First, the original paper where this dataset was published, Riccio et al [51], did not describe the level of cognitive impairment of each participant. In Kellmeyer et al [14], through an extensive review of ALS biomarkers in the context of EEG studies, it was verified that there is evidence of abnormalities of ERP responses and alterations in the waveforms regardless of the level of cognitive impairment scored for each patient. Nonetheless, the influence of any level of cognitive impairment as a possible confounding factor in the ERP response cannot be neglected. Another technical limitation is that this method relies on signal averaging to obtain a meaningful pattern that can be detected by analyzing the shape of the signal. This is reflected in the low success rate in letter identification obtained for only one signal intensification. The application of this method requires that the information is encoded in the waveform of the signal. Concordantly, there is already vast established knowledge in terms of EEG waveform shapes that can be explored further with potential applications of this proposed method.

Author Contributions: Conceptualization, R.R. and J.M.S.; methodology, R.R.; software, B.E.A.; validation, R.R., J.M.S. and J.G.; formal analysis, J.G.; investigation, B.E.A.; resources, J.M.S.; data curation, B.E.A.; writing—original draft preparation, B.E.A.; writing—review and editing, R.R. and J.G.; visualization, B.E.A. and J.G.; supervision, R.R.; project administration, J.M.S.; funding acquisition, J.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Instituto Tecnológico de Buenos Aires (ITBA) grant number ITBACyT-2020. The APC was funded by Instituto Tecnológico de Buenos Aires (ITBA).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this work are the dataset 008-2014 publicly available from BNCI-Horizon 2020 website: <https://bnci-horizon-2020.eu/database/data-sets> (accessed on 19 August 2024).

Acknowledgments: We would like to thank ITBA University for supporting this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Nicolelis, M.A.L. Brain–machine–brain interfaces as the foundation for the next generation of neuroprostheses. *Natl. Sci. Rev.* **2021**, *9*, nwab206. [[CrossRef](#)] [[PubMed](#)]
- Ajiboye, A.B.; Willett, F.R.; Young, D.R.; Memberg, W.D.; Murphy, B.A.; Miller, J.P.; Walter, B.L.; Sweet, J.A.; Hoyen, H.A.; Keith, M.W.; et al. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: A proof-of-concept demonstration. *Lancet* **2017**, *389*, 1821–1830. [[CrossRef](#)] [[PubMed](#)]
- Metzger, S.L.; Liu, J.R.; Moses, D.A.; Dougherty, M.E.; Seaton, M.P.; Littlejohn, K.T.; Chartier, J.; Anumanchipalli, G.K.; Tu-Chan, A.; Ganguly, K.; et al. Generalizable spelling using a speech neuroprosthesis in an individual with severe limb and vocal paralysis. *Nat. Commun.* **2022**, *13*, 6510. [[CrossRef](#)] [[PubMed](#)]
- Willett, F.; Kunz, E.; Fan, C.; Avansino, D.; Wilson, G.; Choi, E.Y.; Kamdar, F.; Hochberg, L.R.; Druckmann, S.; Shenoy, K.V.; et al. A high-performance speech neuroprosthesis. *bioRxiv* **2023**. [[CrossRef](#)] [[PubMed](#)]
- Huggins, J.E.; Krusienski, D.; Vansteensel, M.J.; Valeriani, D.; Thelen, A.; Stavisky, S.; Norton, J.J.; Nijholt, A.; Müller-Putz, G.; Kosmyna, N.; et al. Workshops of the eighth international brain–computer interface meeting: BCIs: The next frontier. *Brain-Comput. Interfaces* **2022**, *9*, 69–101. [[CrossRef](#)]
- Antonietti, A.; Balachandran, P.; Hossaini, A.; Hu, Y.; Valeriani, D. The BCI Glossary: A first proposal for a community review. *Brain-Comput. Interfaces* **2021**, *8*, 42–53. [[CrossRef](#)]
- Orhanbulucu, F.; Latifoğlu, F. Detection of amyotrophic lateral sclerosis disease from event-related potentials using variational mode decomposition method. *Comput. Methods Biomech. Biomed. Eng.* **2022**, *25*, 840–851. [[CrossRef](#)] [[PubMed](#)]
- Pugliese, R.; Sala, R.; Regondi, S.; Beltrami, B.; Lunetta, C. Emerging technologies for management of patients with amyotrophic lateral sclerosis: From telehealth to assistive robotics and neural interfaces. *J. Neurol.* **2022**, *269*, 2910–2921. [[CrossRef](#)]
- Masiello, P. Technology to support autonomy in patients with Amyotrophic Lateral Sclerosis. *J. Adv. Health Care* **2022**, *4*, 47–52. [[CrossRef](#)]
- Vucic, S. P300 jitter latency, brain–computer interface and amyotrophic lateral sclerosis. *Clin. Neurophysiol.* **2021**, *132*, 614–615. [[CrossRef](#)] [[PubMed](#)]
- Guy, V.; Soriani, M.H.; Bruno, M.; Papadopoulo, T.; Desnuelle, C.; Clerc, M. Brain computer interface with the P300 speller: Usability for disabled people with amyotrophic lateral sclerosis. *Ann. Phys. Rehabil. Med.* **2018**, *61*, 5–11. [[CrossRef](#)] [[PubMed](#)]
- Panigutti, C.; Hamon, R.; Hupont, I.; Fernandez Llorca, D.; Fano Yela, D.; Junklewitz, H.; Scalzo, S.; Mazzini, G.; Sanchez, I.; Soler Garrido, J.; et al. The role of explainable AI in the context of the AI Act. In Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, Chicago, IL, USA, 12–15 June 2023; pp. 1139–1150.
- McCane, L.M.; Heckman, S.M.; McFarland, D.J.; Townsend, G.; Mak, J.N.; Sellers, E.W.; Zeitlin, D.; Tenteromano, L.M.; Wolpaw, J.R.; Vaughan, T.M. P300-based brain–computer interface (BCI) event-related potentials (ERPs): People with amyotrophic lateral sclerosis (ALS) vs. age-matched controls. *Clin. Neurophysiol.* **2015**, *126*, 2124–2131. [[CrossRef](#)] [[PubMed](#)]
- Kellmeyer, P.; Grosse-Wentrup, M.; Schulze-Bonhage, A.; Ziemann, U.; Ball, T. Electrophysiological correlates of neurodegeneration in motor and non-motor brain regions in amyotrophic lateral sclerosis—Implications for brain–computer interfacing. *J. Neural Eng.* **2018**, *15*. [[CrossRef](#)] [[PubMed](#)]
- Avola, D.; Cascio, M.; Cinque, L.; Fagioli, A.; Foresti, G.L.; Marini, M.R.; Pannone, D. Analyzing EEG Data with Machine and Deep Learning: A Benchmark. In Proceedings of the Image Analysis and Processing—ICIAP 2022, Lecce, Italy, 23–27 May 2022; Sclaroff, S., Distanto, C., Leo, M., Farinella, G.M., Tombari, F., Eds.; Springer: Cham, Switzerland, 2022; pp. 335–345.
- Guo, J.; Huang, Z. A calibration-free P300 BCI system using an on-line updating classifier based on reinforcement learning. In Proceedings of the 2021 14th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 23–25 October 2021; pp. 1–5. [[CrossRef](#)]
- Khasawneh, N.; Fraiwan, M.; Fraiwan, L. Detection of K-complexes in EEG signals using deep transfer learning and YOLOv3. *Cluster Computing* **2022**, *26*, 3985–3995. [[CrossRef](#)]
- Ramele, R.; Villar, A.J.; Santos, J.M. EEG Waveform Analysis of P300 ERP with Applications to Brain Computer Interfaces. *Brain Sci.* **2018**, *8*, 199. [[CrossRef](#)] [[PubMed](#)]
- Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [[CrossRef](#)] [[PubMed](#)]
- Chen, P.; Gao, Z.; Yin, M.; Wu, J.; Ma, K.; Grebogi, C. Multiattention Adaptation Network for Motor Imagery Recognition. *IEEE Trans. Syst. Man. Cybern. Syst.* **2022**, *52*, 5127–5139. [[CrossRef](#)]
- Kurczak, J.; Białas, K.; Chalupnik, R.; Kedziora, M. Using Brain–Computer Interface (BCI) and Artificial Intelligence for EEG Signal Analysis. In Proceedings of the Recent Challenges in Intelligent Information and Database Systems, Ho Chi Minh City, Vietnam, 28–30 November 2022; Szczerbicki, E., Wojtkiewicz, K., Nguyen, S.V., Pietranik, M., Krótkiewicz, M., Eds.; Springer: Singapore, 2022; pp. 214–226.
- Tabar, Y.R.; Halici, U. A novel deep learning approach for classification of EEG motor imagery signals. *J. Neural Eng.* **2016**, *14*, 016003. [[CrossRef](#)] [[PubMed](#)]
- Zhang, X.; Yao, L.; Wang, X.; Monaghan, J.; McAlpine, D.; Zhang, Y. A survey on deep learning based brain computer interface: Recent advances and new frontiers. *arXiv* **2019**, arXiv:1905.04149.
- Paul, A. Prediction of missing EEG channel waveform using LSTM. In Proceedings of the 2020 4th International Conference on Computational Intelligence and Networks (CINE), Kolkata, India, 27–29 February 2020; pp. 1–6. [[CrossRef](#)]

25. Cai, Q.; Gao, Z.; An, J.; Gao, S.; Grebogi, C. A Graph-Temporal Fused Dual-Input Convolutional Neural Network for Detecting Sleep Stages from EEG Signals. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 777–781. [[CrossRef](#)]
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
27. Li, Y.; Zhang, X.R.; Zhang, B.; Lei, M.Y.; Cui, W.G.; Guo, Y.Z. A Channel-Projection Mixed-Scale Convolutional Neural Network for Motor Imagery EEG Decoding. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 1170–1180. [[CrossRef](#)]
28. Liu, M.; Wu, W.; Gu, Z.; Yu, Z.; Qi, F.; Li, Y. Deep learning based on Batch Normalization for P300 signal detection. *Neurocomputing* **2018**, *275*, 288–297. [[CrossRef](#)]
29. Havaei, P.; Zekri, M.; Mahmoudzadeh, E.; Rabbani, H. An efficient deep learning framework for P300 evoked related potential detection in EEG signal. *Comput. Methods Programs Biomed.* **2023**, *229*, 107324. [[CrossRef](#)] [[PubMed](#)]
30. Kundu, S.; Ari, S. P300 based character recognition using convolutional neural network and support vector machine. *Biomed. Signal Process. Control* **2020**, *55*, 101645. [[CrossRef](#)]
31. Abibullaev, B.; Zollanvari, A. A Systematic Deep Learning Model Selection for P300-Based Brain-Computer Interfaces. *IEEE Trans. Syst. Man, Cybern. Syst.* **2022**, *52*, 2744–2756. [[CrossRef](#)]
32. Singh, S.A.; Meitei, T.G.; Devi, N.D.; Majumder, S. A deep neural network approach for P300 detection-based BCI using single-channel EEG scalogram images. *Phys. Eng. Sci. Med.* **2021**, *44*, 1221–1230. [[CrossRef](#)] [[PubMed](#)]
33. Singh, A.K.; Tao, X. BCINet: An Optimized Convolutional Neural Network for EEG-Based Brain-Computer Interface Applications. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; pp. 582–587. [[CrossRef](#)]
34. Lawhern, V.J.; Solon, A.J.; Waytowich, N.R.; Gordon, S.M.; Hung, C.P.; Lance, B.J. EEGNet: A compact convolutional neural network for EEG-based brain-computer interfaces. *J. Neural Eng.* **2018**, *15*, 056013. [[CrossRef](#)]
35. Zhu, H.; Forenzo, D.; He, B. On The Deep Learning Models for EEG-based Brain-Computer Interface Using Motor Imagery. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 2283–2291. [[CrossRef](#)]
36. Lotte, F.; Bougrain, L.; Cichocki, A.; Clerc, M.; Congedo, M.; Rakotomamonjy, A.; Yger, F. A review of classification algorithms for EEG-based brain-computer interfaces: A 10 year update. *J. Neural Eng.* **2018**, *15*, 031005. [[CrossRef](#)]
37. Alzahab, N.A.; Apollonio, L.; Di Iorio, A.; Alshalak, M.; Iarlori, S.; Ferracuti, F.; Monteriù, A.; Porcaro, C. Hybrid Deep Learning (hDL)-Based Brain-Computer Interface (BCI) Systems: A Systematic Review. *Brain Sci.* **2021**, *11*, 75. [[CrossRef](#)] [[PubMed](#)]
38. Vavoulis, A.; Figueiredo, P.; Vourvopoulos, A. A Review of Online Classification Performance in Motor Imagery-Based Brain-Computer Interfaces for Stroke Neurorehabilitation. *Signals* **2023**, *4*, 73–86. [[CrossRef](#)]
39. Hossain, K.M.; Islam, M.A.; Hossain, S.; Nijholt, A.; Ahad, M.A.R. Status of deep learning for EEG-based brain-computer interface applications. *Front. Comput. Neurosci.* **2023**, *16*, 1006763. [[CrossRef](#)] [[PubMed](#)]
40. Song, W.; Liu, L.; Liu, M.; Wang, W.; Wang, X.; Song, Y. Representation learning with deconvolution for multivariate time series classification and visualization. In Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators, Taiyuan, China, 18–21 September 2020; Springer: Singapore; pp. 310–326.
41. Colyer, A. The way we think about data: Human inspection of black-box ML models; reclaiming ownership of data. *Queue* **2019**, *17*, 26–27. [[CrossRef](#)]
42. Wong, F.; Zheng, E.J.; Valeri, J.A.; Donghia, N.M.; Anahtar, M.N.; Omori, S.; Li, A.; Cubillos-Ruiz, A.; Krishnan, A.; Jin, W.; et al. Discovery of a structural class of antibiotics with explainable deep learning. *Nature* **2023**, *626*, 177–185. [[CrossRef](#)] [[PubMed](#)]
43. Savage, N. Breaking into the black box of artificial intelligence. *Nature*, 29 March 2022.
44. Ail, B.E. EEG Waveform Identification Based on Deep Learning Techniques. Master's Thesis, Instituto Tecnológico de Buenos Aires Buenos Aires, Argentina, 2022.
45. Ramele, R.; Villar, A.J.; Santos, J.M. Histogram of Gradient Orientations of Signal Plots Applied to P300 Detection. *Front. Comput. Neurosci.* **2019**, *13*, 43. [[CrossRef](#)]
46. Ramele, R. Histogram of Gradient Orientations of EEG Signal Plots for Brain Computer Interfaces. Ph.D. Thesis, Instituto Tecnológico de Buenos Aires Buenos Aires, Argentina, 2018.
47. Papastylianou, T.; Dall' Armellina, E.; Grau, V. Orientation-Sensitive Overlap Measures for the Validation of Medical Image Segmentations. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI, Athens, Greece, 17–21 October 2016; Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W., Eds.; Springer: Cham, Switzerland, 2016; pp. 361–369.
48. Ganapathy, K.; Abdul, S.S.; Nursetyo, A.A. Artificial intelligence in neurosciences: A clinician's perspective. *Neurol. India* **2018**, *66*, 934. [[CrossRef](#)]
49. Kawala-Sterniuk, A.; Browarska, N.; Al-Bakri, A.; Pelc, M.; Zygarlicki, J.; Sidikova, M.; Martinek, R.; Gorzelanczyk, E.J. Summary of over Fifty Years with Brain-Computer Interfaces—A Review. *Brain Sci.* **2021**, *11*, 43. [[CrossRef](#)] [[PubMed](#)]
50. Brunner, C.; Birbaumer, N.; Blankertz, B.; Guger, C.; Kübler, A.; Mattia, D.; del R. Millán, J.; Miralles, F.; Nijholt, A.; Opisso, E.; et al. BNCI Horizon 2020: Towards a roadmap for the BCI community. *Brain-Comput. Interfaces* **2015**, *2*, 1–10. [[CrossRef](#)]
51. Riccio, A.; Simione, L.; Schettini, F.; Pizzimenti, A.; Inghilleri, M.; Olivetti Belardinelli, M.; Mattia, D.; Cincotti, F. Attention and P300-based BCI performance in people with amyotrophic lateral sclerosis. *Front. Hum. Neurosci.* **2013**, *7*, 732. [[CrossRef](#)]
52. Schalk, G.; Mcfarland, D.; Hinterberger, T.; Birbaumer, N.; Wolpaw, J. BCI2000: A general-purpose Brain-Computer Interface (BCI) system. *IEEE Trans. Biomed. Eng.* **2004**, *51*, 1034–1043. [[CrossRef](#)] [[PubMed](#)]

53. Abdulaal, M.J.; Casson, A.J.; Gaydecki, P. Performance of Nested vs. Non-Nested SVM Cross-Validation Methods in Visual BCI: Validation Study. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 1680–1684. [\[CrossRef\]](#)
54. Shafer, G.; Vovk, V. A Tutorial on Conformal Prediction. *J. Mach. Learn. Res.* **2008**, *9*, 371–421.
55. Delorme, A. EEG is better left alone. *Sci. Rep.* **2023**, *13*, 2372. [\[CrossRef\]](#) [\[PubMed\]](#)
56. van Drongelen, W. 4-Signal Averaging. In *Signal Processing for Neuroscientists*; van Drongelen, W., Ed.; Academic Press: Burlington, NJ, USA, 2007; pp. 55–70. [\[CrossRef\]](#)
57. Jackson, A.F.; Bolger, D.J. The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. *Psychophysiology* **2014**, *51*, 1061–1071. [\[CrossRef\]](#)
58. Zhang, R.; Xu, P.; Guo, L.; Zhang, Y.; Li, P.; Yao, D. Z-score linear discriminant analysis for EEG based brain-computer interfaces. *PLoS ONE* **2013**, *8*, e74433. [\[CrossRef\]](#) [\[PubMed\]](#)
59. Jestico, J.; Fitch, P.; Gilliatt, R.W.; Willison, R.G. Automatic and rapid visual analysis of sleep stages and epileptic activity. A preliminary report. *Electroencephalogr. Clin. Neurophysiol.* **1977**, *43*, 438–441. [\[CrossRef\]](#)
60. Bresenham, J.E. Algorithm for computer control of a digital plotter. *IBM Syst. J.* **1965**, *4*, 25–30. [\[CrossRef\]](#)
61. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014), 2014. Available online: <https://www.image-net.org/challenges/LSVRC/2014/> (accessed on 3 September 2022).
62. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
63. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
64. Tibon, R.; Levy, D.A. Striking a balance: Analyzing unbalanced event-related potential data. *Front. Psychol.* **2015**, *6*, 555. [\[CrossRef\]](#)
65. Gramfort, A.; Luessi, M.; Larson, E.; Engemann, D.A.; Strohmeier, D.; Brodbeck, C.; Parkkonen, L.; Hämäläinen, M.S. MNE software for processing MEG and EEG data. *NeuroImage* **2014**, *86*, 446–460. [\[CrossRef\]](#)
66. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [\[CrossRef\]](#)
67. Friedman, B. Creating a TensorFlow CNN in C++, 2014. Available online: <https://towardsdatascience.com/creating-a-tensorflow-cnn-in-c-part-2-eea0de9dcada> (accessed on 4 April 2024).
68. Pavlov, Y.G.; Adamian, N.; Appelhoff, S.; Arvaneh, M.; Benwell, C.S.; Beste, C.; Bland, A.R.; Bradford, D.E.; Bublatzky, F.; Busch, N.A.; et al. #EEGManyLabs: Investigating the replicability of influential EEG experiments. *Cortex* **2021**, *144*, 213–229. [\[CrossRef\]](#) [\[PubMed\]](#)
69. Lowe, G. SIFT-the scale invariant feature transform. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
70. Theodorsson-Norheim, E. Friedman and Quade tests: BASIC computer program to perform nonparametric two-way analysis of variance and multiple comparisons on ranks of several related samples. *Comput. Biol. Med.* **1987**, *17*, 85–99. [\[CrossRef\]](#)
71. Roy, Y.; Banville, H.; Carneiro de Albuquerque, I.M.; Gramfort, A.; Falk, T.; Faubert, J. Deep learning-based electroencephalography analysis: A systematic review. *J. Neural Eng.* **2019**, *16*, 051001. [\[CrossRef\]](#)
72. Wolpaw, J.R.; Birbaumer, N.; McFarland, D.J.; Pfurtscheller, G.; Vaughan, T.M. Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* **2002**, *113*, 767–791. [\[CrossRef\]](#) [\[PubMed\]](#)
73. Wolpaw, J.; Wolpaw, E.W., Eds. *Brain-Computer Interfaces: Principles and Practice*; Oxford University Press: Oxford, UK, 2012.
74. Rao, R.P.N. *Brain-Computer Interfacing: An Introduction*; Cambridge University Press: Cambridge, UK, 2013. [\[CrossRef\]](#)
75. Valeriani, D.; Santoro, F.; Ienca, M. The present and future of neural interfaces. *Front. Neurobotics* **2022**, *16*. [\[CrossRef\]](#)
76. Perry, J.C.; Rosen, J.; Burns, S. Upper-Limb Powered Exoskeleton Design. *IEEE/ASME Trans. Mechatron.* **2007**, *12*, 408–417. [\[CrossRef\]](#)
77. Lobo-Prat, J.; Keemink, A.Q.L.; Stienen, A.H.A.; Schouten, A.C.; Veltink, P.H.; Koopman, B.F.J.M. Evaluation of EMG, force and joystick as control interfaces for active arm supports. *J. NeuroEng. Rehabil.* **2014**, *11*, 68. [\[CrossRef\]](#)
78. Coupé, C.; Oh, Y.M.; Dediu, D.; Pellegrino, F. Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Sci. Adv.* **2019**, *5*, eaaw2594. [\[CrossRef\]](#) [\[PubMed\]](#)
79. Kursa, M.B.; Jankowski, A.; Rudnicki, W.R. Boruta—a system for feature selection. *Fundam. Informaticae* **2010**, *101*, 271–285. [\[CrossRef\]](#)
80. Liang, Y.; Li, S.; Yan, C.; Li, M.; Jiang, C. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing* **2021**, *419*, 168–182. [\[CrossRef\]](#)
81. Azam, S.; Montaha, S.; Fahim, K.U.; Rafid, A.K.H.; Mukta, M.S.H.; Jonkman, M. Using feature maps to unpack the CNN Black box theory with two medical datasets of different modality. *Intell. Syst. Appl.* **2023**, *18*, 200233. [\[CrossRef\]](#)
82. Montavon, G.; Samek, W.; Müller, K.R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **2018**, *73*, 1–15. [\[CrossRef\]](#)
83. Linse, K.; Aust, E.; Joos, M.; Hermann, A. Communication matters-pitfalls and promise of hightech communication devices in palliative care of severely physically disabled patients with amyotrophic lateral sclerosis. *Front. Neurol.* **2018**, *9*, 603. [\[CrossRef\]](#)
84. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [\[CrossRef\]](#)

-
85. Ehsan, U.; Wintersberger, P.; Liao, Q.V.; Mara, M.; Streit, M.; Wachter, S.; Riener, A.; Riedl, M.O. Operationalizing Human-Centered Perspectives in Explainable AI. In Proceedings of the Conference on Human Factors in Computing Systems-Proceedings, Online, 8–13 May 2021. [[CrossRef](#)]
 86. Raissi, M. Open Problems in Applied Deep Learning. *arXiv* **2023**, arXiv:2301.11316.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.