University of Padua

Department of Information Engineering

Biomedical Engineering

Explainable deep learning solutions for the artifacts correction of EEG signals

Student:
Federico Donadel

Supervisor:
Dr. Giulia Cisotto

Academic year 2021/2022
September 21th 2022

Ackowledgment

Before proceeding with the discussion, I would like to dedicate a few lines to all those who have been close to me on this years and on this path of study.

A big thanks goes to my relator Cisotto Giulia for her helpfulness and for giving me a lot of confidence,

To my parents Carlo and Monica and my sister Sofia for support me during these years encouraging me to do better and better,

To my girlfriend Anna for always be there, support me, make me happy and put up with me.

To my maternal grandparents, for always give me love, encourage me, and to always be an example for life,

To my paternal grandparents that are watching from above, for make me grow up and gave me good lessons,

To Tisanine & Birrette my group of best friends, for all the good times we spent together,

To Alberto, to be a big help in this work of thesis,

To my colleagues,

To all my family and friends,

Abstract

The brain electrical activity can be acquired via electroencephalography (EEG) with electrodes placed on the scalp of the individual. When EEG signals are recorded, signal artifacts such as muscular activities, blinking of eyes, and power line electrical noise can significantly affect the quality of the EEG signals [1]. Machine learning (ML) techniques are an example of method used to classify and remove EEG artifacts. Deep learning is a type of ML inspired by the architecture of the cerebral cortex, that is formed by a dense network of neurons, simple processing units in our brain. [2]. In this thesis work we use ICLabel [3] that is an artificial neural network developed by EEGLAB to automatically classify, that classifies the inidpendent component(ICs), obtained by the application of the independent component analysis (ICA), in seven classes, i.e., brain, eye, muscle, heart, channel noise, line noise, other. ICLabel provides the probability that each IC features belongs to one out of 6 artefact classes, or it is a pure brain component. We create a simple CNN similar to the ICLabel's one that classifies the EEG artifacts ICs in two classes, brain and not brain. and we added an explainability tool, i.e., GradCAM, to investigate how the algorithm is able to successfully classify the ICs. We compared the performances f our simple CNN versus those of ICLabel, finding that CNN is able to reach satisfactory accuracies (over two classes, i.e., brain/non-brain). Then we applied GradCAM to the CNN to understand what are the most important parts of the spectrogram that the network used to classify the data and we could speculate that, as expected, the CNN is driven by components such as the power line noise (50 Hz and higher harmonics) to identify non-brain components, while it focuses on the range 1-30 Hz to identify brain components. Although promising, these results need further investigations. Moreover, GradCAM could be later applied to ICLabel, too, in order to explain the more sophisticated DL model with 7 classes.

Abstract

L'attività celebrale può essere acquisita tramite elettroencefalografia (EEG) con degli elettrodi posti sullo scalpo del soggetto. Quando un segnale EEG viene acquisito si formano degli artefatti dovuti a: movimenti dei muscoli, movimenti degli occhi, attività del cuore o dovuti all'apparecchio di acquisizione stesso. Questi artefatti possono notevolmente compromettere la qualità dei segnali EEG. La rimozione di questi artefatti è fondamentale per molte discipline per ottenere un segnale pultio e poterlo utilizzare nel miglire dei modi. Il machine learning (ML) è un esempio di tecnica che può essere utilizzata per classificare e rimuovere gli artefatti dai segnali EEG. Il deep leraning (DL) è una branca del ML che è sviluppata ispirandosi all'architettura della corteccia cerebrale umana. Il DL è alla base della creazione dell'intelligenza artificiale e della costruzione di reti neurali (NN) [2]. Nella tesi applicheremo ICLabel [3] che è una rete neurale che classifica le componenti indipendenti (IC), ottenute con la scomposizione tramite independent component analysis (ICA), in sette classi differenti: brain, eye, muscle, heart, channel noise, line noise e other. ICLabel calcola la probabilità che le ICs appartengano a ciascuna di queste sette classi. Durante questo lavoro di tesi abbiamo sviluppato una semplice rete neurale, simile a quella di ICLabel, che classifica le ICs in due classi: una contenente le ICs che corrispondono a quelli che sono i segnali base dell'attività cerebrale, l'altra invece contenente le ICs che non appartengono a questi segnali base. Abbiamo creato questa rete neurale per poter applicare poi un algoritmo di explainability (basato sulle reti neurali), chiamato GradCAM. Abbiamo, poi, comparato le performances di ICLabel e della rete neurale da noi sviluppata per vedere le differenze dal punto di vista della accuratezza e della precisione nella classificazione, come descritto nel capitolo 4. Abbiamo infine applicato gradCAM alla rete neurale da noi sviluppata per capire quali parti del segnale la rete usa per compiere le classificazioni, evidenziando sugli spettrogrammi delle ICs le parti più importanti del segnale. Possiamo dire poi, che come ci aspettavamo la CNN è guidata da componenti come quelle del line noise (che corrisponde alla frequenza di 50 Hz e armoniche più alte) per identificare le componenti non brain, mentre si concentra sul range da 1-30 Hz per identificare quelle brain. Anche se promettenti questi risultati vannno investigati. Inoltre GradCAM potrebbe essere applicato anche su ICLabel per spiegare la sua struttura più complessa.

Outline

The brain electrical activity can be acquired via electroencephalography (EEG) with electrodes placed on the scalp of the individual. When EEG signals are recorded, signal artifacts such as muscular activities, blinking of eyes, and power line electrical noise can significantly affect the quality of the EEG signals [1]. These artifacts are hard to eliminate due to their frequencies range that are nearly the same of the EEG bands. Detection, correction, or removal of physiological/internal and non-physiological/external artifacts is an important process to minimize the chance of misinterpretation of EEG, not only for clinical and non-clinical fields such as brain computer interface (BCI), but also for intelligent control system, robotics, etc. [4]. For these reasons, several methods have been developed to classify and remove the artifacts from EEG signals. Generally it is applied the independent component analysis (ICA) to the multichannel EEG signals, before the classification. This method separates multichannel EEG signal mixtures into a corresponding set of statistically independent components (IC). These ICs are manually selected with a great deal of time, and also, with the necessity of a clinical expertise. To speed up and simplify the problem, several automatic methods were developed to classify an remove EEG artifacts.

Machine learning (ML) techniques are an example of method used to classify and remove EEG artifacts. Deep learning is a type of ML inspired by the architecture of the cerebral cortex, is foundational for building artificial neural network [2]. One deep learning feedforward network easier to train is convolutional neural networks (CNN). The work of Pion-Tonachini et al. [3] presents a new CNN-based independent components (ICs) classifier. The algorithm is called IClabel and is an artificial neural network, that classifies the ICs in seven classes, i.e., brain, eye, muscle, heart, channel noise, line noise, other. It provides the probability that the IC features belong to these classes. To do so, each IC is described by its spatial importance (i.e., topography), its time course and its power spectral density (PSD). These data are used as inputs for the CNN-based classifier.

As all neural networks, ICLabel is unable to explain the decision it takes to the users, and is considered a black box. For these reasons understand how the algorithm takes decisions could be very interesting to make the most of its capabilities. To this aim, an explainability algorithm 1.4 can be used. Particularly, in this thesis work, we selected GradCAM 3.3, as it is good in performance, demonstrated when applied to CNN architectures.

The aim of the thesis should have been to apply GradCAM, to ICLabel to understand better how the CNN algorithm classifies the EEG artifacts into seven different classes,

highlighting the regions of the IC spectrogram that are more important. However, ICLabel was developed with Python by Pion-Tonachini and colleagues, then imported to MATLAB and used as EEGLAB plug-in. Unfortunately, ICLabel has been developed using some ad-hoc functions that was not possible to easily use in conjunction with GradCAM. Therefore, we decided to implement a simple CNN similar to the ICLabel's net. We compared the performances between ICLabel and the best results of the CNN's training, as described in the chapter 4, to see how far the performances of the CNN are from those of ICLabel. We also applied GradCAM to the CNN to understand what are the most important parts of the spectrogram that the net uses to classify the data.

Then, the main contributions of this thesi work are:

- the development of a new simple CNN implemented in MATLAB. The training, validation and test of the abovementioned CNN on the same dataset used in a previous MSc thesis work [5].
- The comparison the classification performance between the simple CNN (using 2 classes, i.e., "brain" vs "non-brain") and ICLabel.
- The application of GradCAM to the CNN.

The first chapter offers a brief overview of the thesis work, enlightening the context, the motivation and the main objective. The second chapter, 1, Background, is dedicated to the theoretical background needed for understanding the thesis. Chapter 2, State of art, describes the state of the art of the methods of explainability used to explain the decisions made by deep learning-based classifiers operating on a corrupted EEG dataset (with artefacts). Chapter 3, Methods describes the the algorithms implemented and used on this thesis work. Finally chapter 4, Results and discussion, presents the main results and the conclusions of this thesis work.

Contents

List of Figures

Chapter 1

Background

1.1 Electroencephalography

EEG is the physiological noninvasive method to record the electrical activity generated by the brain with electrodes placed on the scalp surface of the individual [6]. EEG measures the electric potentials generated by the neurons. Between 10 to 20 billion neurons are contained only in the gray matter [10]. The neurons are electrical excitable cell that communicate with each other by connection called synapses. Neurons are composed by a cell body, called soma, dendrites and a single axon. The soma contains the nucleus of the cell and is surrounded by dendrites, that connect the cell with other neurons, and an axon. The axon is a cable-like projection that can extend tens, hundreds, or even tens of thousands of times the diameter of the soma in length. Its function is to carry the nerve signals that is an all-or-nothing electrochemical pulse called *action potential*. The action potential runs along the axon thanks to ions exchange between the inside and the outside of the membrane. To keep rapid conduction, neurons have insulating sheaths of myelin around their axons, formed by glial cells, enables action potentials to travel faster. The hall where the myelin is punctuated are called nodes of Ranvier, which contain a high density of voltage-gated ion channels [11]. In the synapses, at the end of the dendrites and of the axon, the signals cross from the axon of one neuron to a dendrite of another, through neurotransmitters. Figure 1.1.

Electrical potentials are created by the postsynaptic potentials generated at apical dendrites of pyramidal cells in the cortex. The poles of the electric dipole can be seen as the source of a ionic currents created by the excess and defect of cations in the soma and apical dendrites, respectively. These ions can freely move through the cerebrospinal fluid and brain tissues, thus causing ionic currents [12]. EEG provides excellent time resolution (in the order of ms), allowing to detect activity in different cortical areas in real-time with very complex tasks.

The cortex is divided into four areas as in Figure 1.2 and they are associated to different processing of the information:

1. Occipital cortex: it is the visual processing center of our brain, located in the rear

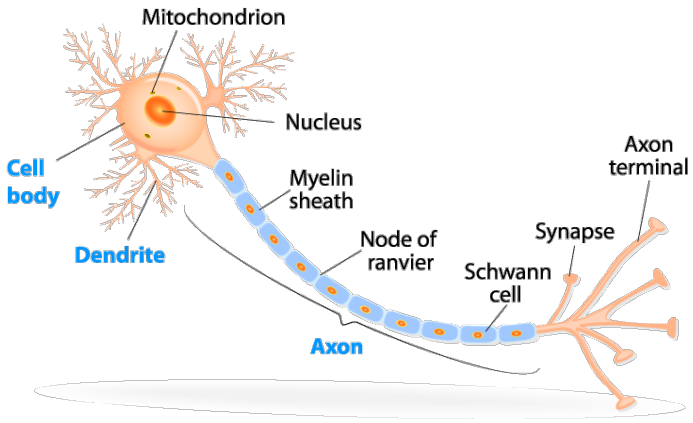

Figure 1.1: Structure of a typical neuron.

portion of the skull.

2. Parietal cortex: it integrates information stemming from external sources and internal sensory feedback from our body.

3. Temporal cortex:it is associated to the processing sensory input to derived, meanings, using visual memories, language and emotional associations.

4. Frontal cortex: it helps us maintain control, plans for the future, and to monitor our behavior.

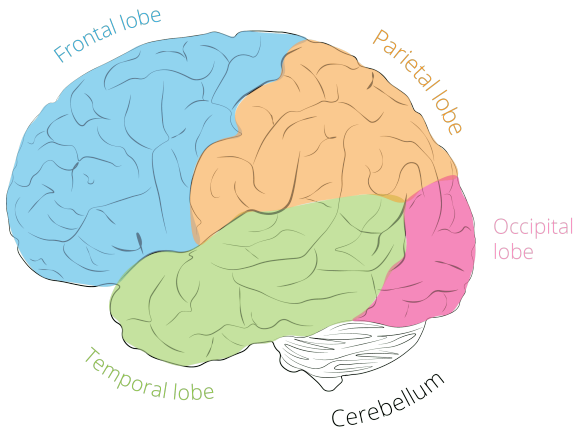

Figure 1.2: cortical areas. From [6]

The multi-channel EEG recordings are always a mixture of several underlying base frequencies, which are considered to reflect certain cognitive, affective, or attentional states [6]. These frequencies are associated with different types of waves that represent several states of mind Figure 1.4:

- Delta (δ) waves with a frequency of range $[0.1 - 4]$Hz. In sleep labs, delta waves are examined to assess the depth of sleep.

- Theta (θ) waves with a frequency of range $[4-8]$Hz, are associated with a wide range of cognitive processing such as memory encoding and retrieval as well as cognitive workload.

- Alpha (α) waves with a frequency of range $[8-13]$Hz, are associated to relax wakefulness of the individual.

- Beta (β) waves with a frequency of range $[13-20]$Hz. Over motor regions, beta frequencies become stronger with the execution of movements of any body part.

- Gamma (γ) waves with a frequency of range $[30-100]$Hz, Some researchers argue that gamma reflects attentive focusing and serves as carrier frequency to facilitate data exchange between brain regions [13]

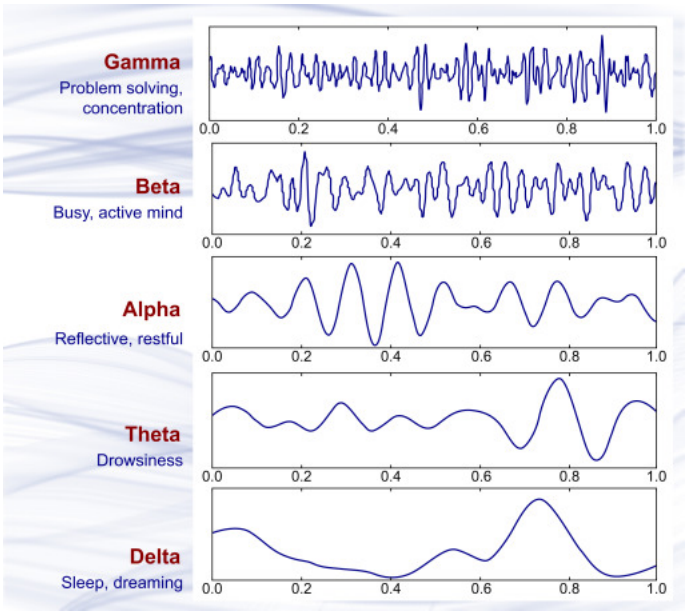

Figure 1.3: Brain waves

1.1.1 Electrodes placement on the scalp

To measure the EEG signals are used electrodes placed on the scalp. There's different standard system suitable for describing the locations of scalp potential measurements, the first, proposed by Jasper in 1958 [14] is the 10-20 system with 21 electrodes. After, in 1985 Chatrian [15] proposed the 10-10 method with 74 electrodes. In 2000 another system was proposed, with 345 electrodes, called 10-5 system [16]. These electrodes are labeled by letters (i.e. F-Frontal, O-Occipital, T-Temporal, C-Central, P-Parietal) which indicate the lobes. Odd numbers indicate the left hemisphere and even numbers indicate the right hemisphere. The electrodes are also placed in two methods (i.e. montages) depending on whether are referential or bipolar electrodes. In referential method, each electrode records

the potential differences compared to a reference electrode, placed on both ear lobes. In bipolar method the potential is recorded between paired active electrodes [17].

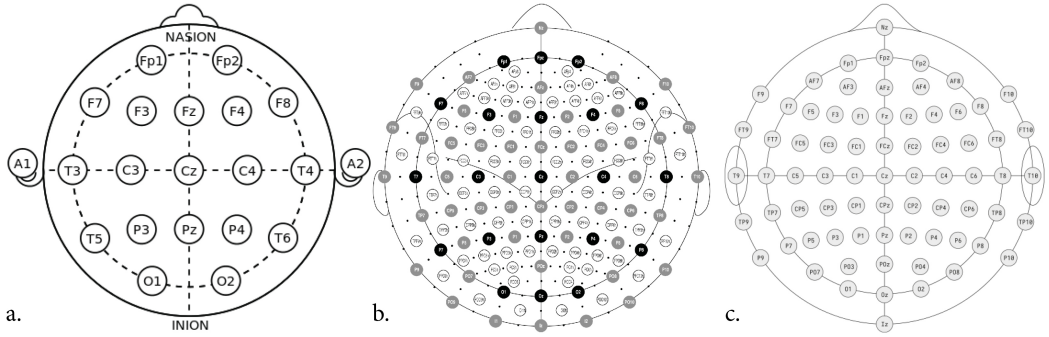

Figure 1.4: Location of the electrodes in the scalp: a. 10-20 system, b. 10-5 system, c. 10-10 system. Figures modified from [7]

1.2 Artefacts and interferences in EEG

When EEG signals are recorded, signal artifacts such as muscular activities, blinking of eyes, and power line electrical noise can affect the quality of the EEG signal [1]. For this reason there are different methods to classify, recognise and eliminate EEG artifacts.

There are several types of artifacts and they may be broadly categorized as physiological/internal and non-physiological/external artifacts. Physiological artifacts are related to physiological sources of the human body like ocular artifacts, muscle artifacts, and cardiac artifacts [18]. The non-physiological artifacts are related to external factors and include: power line noise (50/60 Hz), electrode malfunction (loose connection with the scalp or high impedance electrodes), electromagnetic interference, and variations of the impedances [18].

1.2.1 Ocular artifacts

Electroculogram (EOG) is major source of contamination of EEG. The EOG measures, the electrical activity produced by eye movement, are primarily picked up by the frontal electrodes, and the amplitude is generally much larger than that of the background EEG activity [19].

1.2.2 Muscular artifacts

The electromyogram (EMG) measures the electrical activity on the body surface caused by contracting muscles. This artifact is typical of patients who are awake and occurs when the patient swallows, talks, walks, etc [20]

1.2.3 Cardiac artifacts

Interference due to cardiac activity (either electrical or pulse related) generates a periodic waveform at a frequency that corresponds to the cardiac frequency (expressed in beats per minute, BPM). Typically at rest, cardiac frequencies vary from 60 to 100 BPM [21] for healthy individuals and from 40 to 50 BPM for athletes [22].

1.2.4 Other artifacts

Other possible artifacts include movements of the tongue, clenching of the teeth, breathing artifacts in the lower part of the spectrum, electrodermal interferences due to sweating, chest movements, etc. [23].

1.3 Convolutional neural network (CNN)

In this thesis the type of neural network used is CNN [24]. CNNs could be composed by several types of layers, the principals are: convolutional layers, pooling layers, fully-connected layers and classification layers, that stack together form the CNN architecture. As explained by O'Shea et al. [25]: the convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume, this layer also reduces the complexity of the network thanks to three hyperparameters: depth, stride and zero-padding. The pooling layer simply performs downsampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation. The fully-connected layer perform the same duties of a standard neural network and attempts to produce class scores from the activations, to be used for classification. Learnable kernels are fundamental to the creation of 2D activation map. The scalar product is calculated for each value in that kernel and the network learn kernels that 'fire' when they see a specific feature at a given spatial position of the input. These are commonly known as activations.

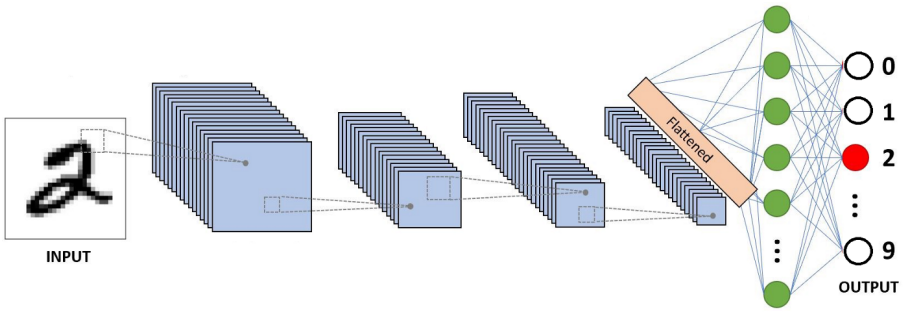

Figure 1.5: An example of CNN structure.(From [8])

1.3.1 Deep Learning

CNN is a neural network based on deep learning. As explained by Dong review [26], Deep learning is a training algorithm with layer-by-layer unsupervised learning and training. It's composed by many classifiers working together, which are based on linear regression followed by some activation functions. The statistical linear regression approach is composed by only one node, and one classifier node is known as a neural unit or perception. Every layer can have many hundreds or even thousands of neural units. The layers which are in between the input and the output are known as the hidden layers, and the nodes are known as the hidden node Figure 1.7 . The activation functions (e.g. Relu, Sigmoid, etc.) are used to generate nonlinear relationships between the input and the output, and to transform and abstract the data into a more classifiable plane. Non linearity, combined with many neural nodes and many layers, mimics the human brain like structure, which is why it is called a neural network (NN). Technique as drop-out, that switch off some of the neural units randomly, are used to prevent over-fitting.

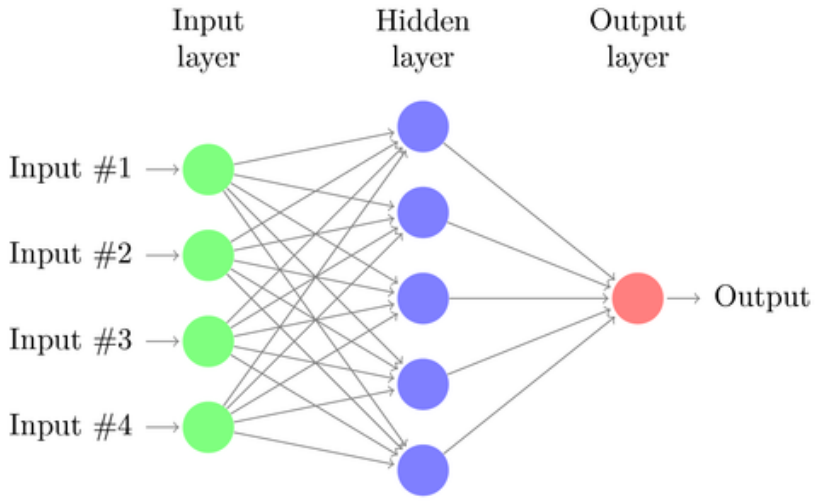

Figure 1.6: General Artificial Neural Network structure (from Saha et al. [27]).

Dong et al. [26] survey the deep learning-based models that mainly includes: stacked automatic encoder,deep belief network, deep Boltzmann machine, convolution neural network etc. The automatic encoder mainly consists of the encoder, the decoder and the hidden layer, staking automatic encoder is an automatic encoder upgraded, restricted Boltzmann machine (RBM) can be represented by a network of stochastic binary neurons, whose states are observable, which are connected to stochastic, unobservable, hidden units [28]. Deep belief network (DBN) is composed of a superposition of multiple constrained Boltzmann models with hidden explanatory factor and neural networks of multiple layers. Deep Boltzmann Machine (DBM) is also formed by the restricted Boltzmann machine stack, which is similar to the deep belief network, with the differences that they have the former layer and the current layer between the non directional connections, and there are no feedback parameters from top to bottom.

1.4 Explainability

Despite the good performances of the EEG automated classifier, as the general purpose nature, and the ability to merge together features extraction and classification, these algorithms, are not able to explain and demonstrate their decisions to human users. For these reasons they are considered black boxes. Understand the decision procedures of the algorithm is essential, expecially in some applications as health sector [29].

Sometimes, the highest performing methods (e.g., DL) are the least explainable, and the most explainable (e.g., decision trees) are the least accurate [30]. In the last years have been developed new artificial intelligence (AI) methods to explain and control the algorithm's decisions: this branch of AI is called explainable AI (XAI). Among the DL methods for the explainability there are: *Local Interpretable Model-agnostic Explanations (LIME)*, *SHapley Additive exPlanation (SHAP)*, *attention mechanism* and the *class activation mapping (CAM)*. LIME concretely tests how predictions change when data variations are given to the model, LIME could be used to interpret the black box of neural networks, but also to interpret boosting-based method as NGBoost as cited by Barus et al. [31]. SHAP falls into class of additive feature attribution methods and is associated to Shapley values, a game theory concept that is based on the attribution of certain importance values to the input features. SHAP values are consistent, even when the order in which features appear in the tree changes, for this reason as described in the article of Lundberg and Lee [32], SHAP values provide a strict theoretical improvement over existing approaches by eliminating the un-intuitive consistency problems. SHAP respects all the three desirable properties that the additive feature attribution methods class have [32]: local accuracy, missingness, and consistency. Attention mechanisms are components of prediction systems that allow the system to sequentially focus on different subsets of the input. The selection of the subset is typically conditioned on the state of the system which is itself a function of the previously attended subsets [33]. The CAM are attention techniques used to identify a sort of "heat map" highlighting regions that support the classification by the CNN into specific categories [34]. As described in section 3.3 GradCAM is an example of a specific CAM implementation that use class-specific gradient information to localize important regions of the data.

Layer-wise relevance propagation (LRP) [35] is another algorithms for image classifications, the core idea underlying the LRP algorithm, for attributing relevance to individual input nodes, is to trace back contributions to the final output node layer by layer. LRP applies a propagation rule that distributes class relevance found at a given layer onto the previous layer. The layer-wise propagation rule was applied iteratively from the output back to the input, thus, forming a possible pixelwise decomposition. It utilizes the gradients and activations of the network to estimate relevance, outputting both positive and negative relevance indicating the features that provide evidence of a sample being assigned to a class (for positive relevance), or in contrast indicates features that provide evidence for a sample being assigned to classes other than what it is ultimately assigned to by the

classifier (for negative relevance) [36]. This inherits the favorable scaling properties of backpropagation.

1.5 Rejection methods

Rashmi and colleagues [37] and Gorjan et al. [38] present an overview of the available methods to classify and remove artifacts. Some precautionary measures followed during EEG signals acquisition can prevent some artifacts, such as avoid eye blinks and movement as much as possible, but this cannot be a practical solution for all the applications. Artifacts detection is the most important step for handling artifacts because, often, the artifacts overlap with EEG signals in both spectral and temporal domains and it becomes difficult to use simple filtering or straight forward signal processing technique to identify these artifacts [39]. Artifacts segment rejection methods remove the segment or channel which causes the artifact, but it also eliminates the important neural activities. For this reason the artifacts removal methods, such as regression methods, filters or decomposition techniques, are used to eliminate or correct the artifact without affecting the characteristics of raw signal. The artifacts removal methods include: single artifact removal methods, blind source separation (BSS), empirical mode decomposition (EMD), filters, signal space projection (SSP), beamforming, artifact subspace reconstruction (ASR), and hybrid methods [39].

Several methods have been developed to classify and remove the artifacts from EEG signals [37] [38]. Conventional methods to detect and remove artifacts employ, e.g., filters, technologist visually inspects the data and remove artifacts-affected slots, called manual methods, and automatic methods, that use mathematical algorithms.

1.5.1 Manual rejection

Manual rejection is a common practice in the BCI field. Trials are visually checked by an expert, and those contaminated with artifacts are removed from the analysis it also has the advantage of not being computationally demanding, as it is assumed that a human expert has identified all the artifact-contaminated epochs. The drawbacks of this method are: time-expensiveness and technical expertise, the process of selecting the artifact-free trials may become subjective, and the eventual loss of data due to offline analysis [40].

1.5.2 Spatial filters

Adaptive filtering
It can be used to remove the physiological artifacts using them as the reference signal. The weights are updated iteratively to subtract the artifact from the raw signal.
Low and high-pass filters
They are commonly used before other artifact removal methods, only if the frequency bands of artifacts are known and do not overlap the EEG signals.
Bayesian filters

Bayesian filters use the recorded signal to estimate the EEG state based on the probability. It then use a prediction-correction technique with two models: time update model, describes how the state updates from one time point to another, measurement model, that describes how the recorded data relates to the internal state of the brain [41].

1.5.3 Single artifact removal methods

Regression model

This method uses one or more reference channel to identify and remove the artifacts as electrooculography (EOG), it's composed by a linear equation with two transmission coefficients that correlate EEG and EOG. The main problem of this method is that it fails when there is no reference channel [42].

Wavelet transform

Wavelet decomposition can be used to remove the artifacts from EEG signal using detailed and approximation coefficients with thresholding. The detailed coefficient is given by high pass filter and approximation coefficient is given by low pass filter. The drawback of this method is that it cannot identify the artifacts when they are overlapped with the spectral features [43].

1.5.4 Indipendent Component Analysis

Indipendent Component Analysis (ICA) is a method for decomposing data. ICA separates a set of signal mixtures into a corresponding set of statistically independent components (IC) [44]. ICA belongs to a class of blind source separation (BSS) methods for separating data into underlying informational components. BSS is the most popular method of artifact removal. It considers the EEG signals recorded and the original signals as mixing matrix and gets the estimated sources of artifacts. Very little is known about the recorded signals, and that is why these methods are called 'blind' [45]. To this class belongs: ICA, Canonical correlation analysis (CCA) that is an automatic method that reduces the computational time due to second-order statistics method to fetch the components from uncorrelated feature [46], Principal component analysis (PCA) that is used to construct the mixing matrix based on normalized Eigen-vectors of covariance matrix, coefficients are sorted, based on the first largest value of variance which makes them orthogonal [47], and MCA is limited to few artifacts. It decomposes the signal depending on the morphology of EEG signal whose is already stored in the database [48].

ICA is considered as an extension of the PCA technique. However, PCA optimizes the covariance matrix of the data which represents second-order statistics, while ICA optimizes higher-order statistics such as kurtosis. The main goal of this algorithm is to extract independent components by maximizing the non-Gaussianity, minimizing the mutual information, or using maximum likelihood (ML) estimation method [49]. Considering the spatial separation of the EEG signals, each sensor records a different mixture of the sources,

modelling the mixing process with matrix multiplication the simple model equations are:

$$x(t) = \mathbf{As(t)} \tag{1.1}$$

Where **A** is an unknown matrix called the *mixing matrix* and $\mathbf{x(t)}$, $\mathbf{s(t)}$ are the two vectors representing the observed signals and source signals respectively. The objective is to recover the original signals, $\mathbf{s_i(t)}$ from only the observed vector $\mathbf{x_i(t)}$. We obtain estimates for the sources by first obtaining the "unmixing matrix" **W**, where, $\mathbf{W} = \mathbf{A^{-1}}$. This enables an estimate, $\mathbf{\hat{s}(t)}$, of the independent sources to be obtained:

$$\mathbf{\hat{s}(t)} = \mathbf{Wx(t)} \tag{1.2}$$

The above descriptions are a generalization of the ICA decomposition [50], that takes into account three fundamental assumptions: the sources being considered are statistically independent, the independent components have non-Gaussian distribution and, the mixing matrix is invertible. These assumption allow to decomposed the signals into ICs, knowing only very little information about the mixing process and about the sources themselves.

1.5.5 Empirical mode decomposition (EMD)

EMD is used for non-stationary, non-linear signal processing. It decomposes the signal using fractional gaussian noise (fGn) that removes artifacts using data adaptive detrending approach. The basis of decomposition in this method is intrinsic mode function (IMF) which are finite set of amplitude modulation (AM)-frequency modulation (FM) oscillating components [51].

1.5.6 Signal space projection (SSP)

In the SSP methods, signals with stable spatial patterns are separated into set of components in multidimensional space but the amplitude varies depending on time. It works on the assumption that subspace of the neural signal is different or orthogonal compared to artifact signal [52].

1.5.7 Hybrid methods

There are many hybrid methods proposed by few researchers to overcome the limitations of single artifact removal methods: *Adaptive filtering and BSS*, *Wavelet and BSS*, *EMD and BSS*, *BSS and support vector machine (SVM)*, and others.

Artifact detection or removal is also addressed using machine learning and deep learning models, considering it as hybrid methods. SVM combined with other removal methods is the most used hybrid method as described by Rashmi et al. [37], followed by DL and ANN. Also K-nearest neighbor (KNN), decision tree, linear regression, Bayesian model, and Bagged tree ensemble model, are used, but less frequently.

Many MATLAB/Pythons toolboxes have been recently developed to implement the most common algorithms to detect and remove artefacts as described in table 1.1

1.6 Artifact subspace reconstruction (ASR)

ASR is an automatic, online-capable, component-based method that can remove transient or large-amplitude artifacts. ASR is similar to principal component analysis (PCA)-based method in which large-variance components are rejected and channel data are reconstructed from remaining components [53] with the difference that ASR automatically identifies and utilizes clean portions of data to determine thresholds for rejecting components. Chang et al. [54] studies showed that ASR cleaning improved the quality of a subsequent ICA decompositions, and that this method is particularly effectives for eye and muscle artefacts cancellation.

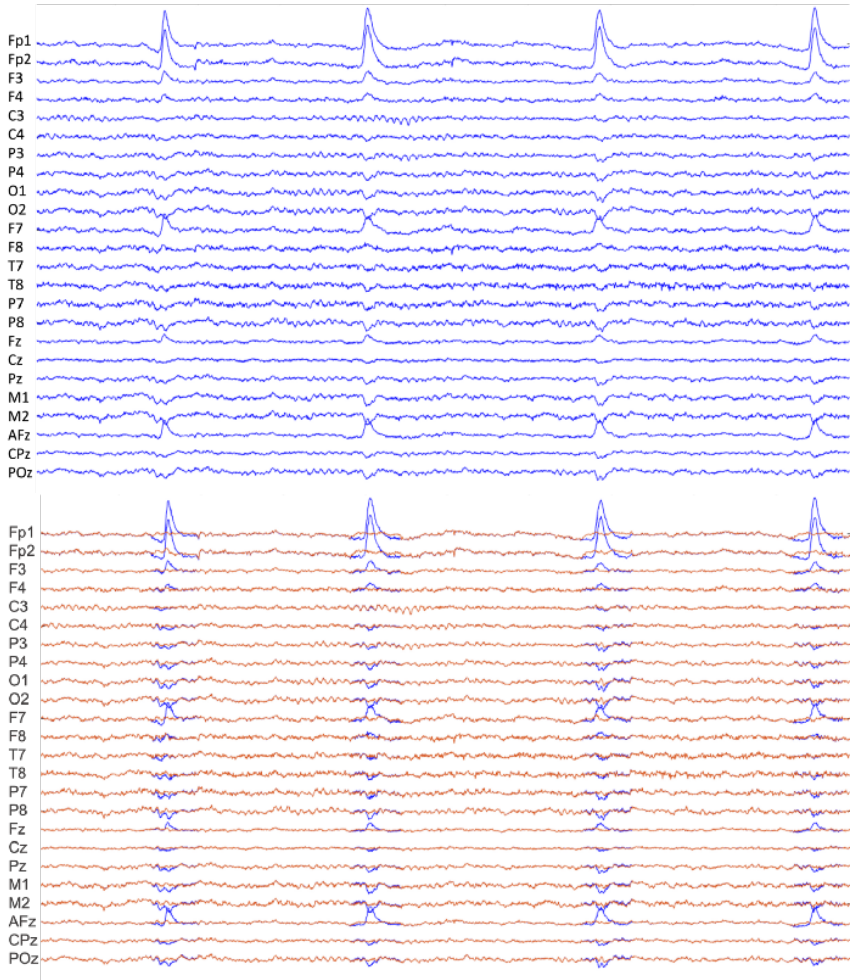

Figure 1.7: Example of EEG eye-blinking artefact cleaned with ASR (from [9]).

Toolbox	**Year**	**Authors**	**Performances**	**Types of artefacts**	**Techniques**
ReMAE	2020	Chen et al. [55]		Muscle	All state of the art methods
ICLabel	2019	Pion-Tonachini et al. [3]	Outperform IC_MARC	All	ICA CL-LDA ASR GAN-based classifiers
HEAR	2019	Kobler et al. [56]		High-variance electrode artifacts	Detection depends on electrode variance
IC_MARC	2015	Frolich et al. [57]	Outperform ADJUST, MARA and FASTER	All	Multinomial regression
SASICA	2015	Chaumon et al. [58]		All	ICA
MNE	2014	Gramfort et al. [59]		All	ICA Connectivity Analysis Statistical Analysis Python Implementation of Pre-Processing Pipeline Automatic Bad Channel Detection and Interpolation
FORCe	2014	Daly et al. [60]	Outperform LAMIC and FASTER	Eye movement, movement, ECG and EMG	Wavelet decomposition with ICA
EEGLAB	2013	Brunner et al. [61]		All	ICA Artifact rjection Filtering Time/Frequency Analysis Event-Related Statistics Visualizations
ADJUST	2011	Mognon et al. [62]		Artifacts from ERP	ICA
FieldTrip	2011	Oostenveld et al. [63]		MEG and EOG	Time-Frequency Analysis Source Reconstruction
MARA	2011	Winkler et al. [64]		All	ICA Supervised learning
FASTER	2010	Nolan et al. [65]		All	Artifact rejection based on ICA
CORRMAP	2009	Viola et al. [66]		Eye movement and heartbeat	Based on ICA
LAMIC	2007	Nicolaou et al. [67]		Artifacts for ERP	Uses BSS with ICA. Followed by clustering.

Table 1.1: Table of Matlab and Python plugins and toolboxes

Chapter 2

State of art

In the recent years many methods of explainability were applied to EEG classifiers. As said in the background 1.4, the automated classifiers such as ICLabel merge together features extraction and classification and have a general purpose nature. They are based on deep learning and neural network algorithms and for these reasons they are not able to explain and demonstrate their decisions to human users. The XAI helps the users to explain the decision making procedures of the algorithms highligghting those parts of the data/features that are important for the results of the classifications.

After a research on Google scholar on the attention methods applied for the EEG artifacts classification and removal with key words as "attention & EEG artifacts",or "deep learning EEG classification artifacts & attention or CNN & attention", I found that quite all the papers and articles applied the attention methods to classify the artifacts, but not to explain how the algorithm of classification works. Searching for "explainability algorithms & artifacts EEG classifier", or "EEG classifier & and explainability", the results are more exhaustive.

The results obtained highlights that, at the state of the art, LRP [35] and GradCAM are two of the most used algorithms for explainability. LRP is a gradient-based feature attribution approach. The core idea underlying the LRP algorithm, for attributing relevance to individual input nodes, is to trace back contributions to the final output node layer by layer. It utilizes the gradients and activations of the network to estimate relevance, outputting both positive and negative relevance indicating the features that provide evidence of a sample being assigned to a class (for positive relevance), or in contrast indicates features that provide evidence for a sample being assigned to classes other than what it is ultimately assigned to by the classifier (for negative relevance) [36]. GradCAM, as described in the section 3.3, is a specific CAM implementation that use class-specific gradient information to localize important regions of the images, in general, which supported the classification as favorable outcome.

In the last six years several applications of these algorithms were developed for various contexts. Ellis and colleagues [36], for example, applied ablation and LRP to evaluate the approach within the context of automated sleep stage classification and find that, for the

most part, the explainability results are highly consistent with clinical guidelines. Other researchers in the past years used LRP. For example, Sturm et al. [68] applied LRP to solve a DNN classification task related to motor-imaginary BCI. They used LRP to produce heatmaps that indicate the relevance of each data point of a spatio-temporal EEG epoch for the classifier's decision in single trial. Also GradCAM is widely used as algorithm for explainability. Li and colleagues [69], for example, applied GradCAM visualization technology to the EEG channel selection for a BCI application. In particular they used a recurrent-CNN structure for EEG intention recognition [69], preprocessing preserves, and captures spatial information by converting the original one-dimensional EEG data vector into a two-dimensional EEG data matrix. Then they applied GradCAM for the channel selection. One other example of application of an algorithm for explainability is given by Ye et al. [70]. In their work they evaluate the performance of a three-dimensional joint convolutional and recurrent neural network for the detection of intracranial hemorrhages in non-contrast head CT. They applied GradCAM to highlight important regions in the image leading to the decision of the algorithm. Comparing the important areas in the heat map and the bleeding positions in the CT images, the algorithm localize in every slice the positions of the bleeding areas and predicts the correct type of intracranial hemorrhage. Also Chen and colleagues [71] and Jonas and colleagues [72] implemented GradCAM to identify the class-discriminative region of the feature maps, and the EEG features that the two CNN algorithms derive. Respectively, they detect abnormalities in EEGs of children with ADHD and predict the clinical outcome in comatose patients after cardiac attack.

GradCAM was, also, tested against other algorithms for explainability, as LRP, as shown by Arias-Duart and colleagues [73]. They compare some popular explainability techniques such as LIME or SHAP, across several CNN architectures and classification EEG datasets. Looking at performances of GradCAM, Arias-Duart and collegues demonstrates the best results on average, and robustness to noisy models. LRP, instead, performs very well in high accuracy models, outperforming GradCAM, but on other less accurate models LRP does not even reach the average results of GradCAM. Also LIME performs remarkably well for high accuracy model but for lower accuracy models it becomes less reliable. For the good performances demonstrated when applied to CNN architectures, and against other explainability algorithms we decided to use GradCAM to highlight the region of the data that our implemented CNN uses to classify the EEG artifacts.

Chapter 3

Methods

In this chapter we firstly discuss ICLabel, an EEGLAB plug-in, in the section 3.1, that we compare against our implemented CNN, discussed in section 3.4. It's a simple neural network, that I have created in this thesis work, similar to the net of ICLabel. We also applied GradCAM, discussed in section 3.3, to the CNN implemented to understand what are the most important regions of the data that the net uses to classify the EEG artifacts. In the section 3.2 we discuss about short-time Fourier transform we used to represent the time-frequency images of the ICs contained in the dataset. Instead, the dataset is discussed in section 3.6. All the details about the implementation are given in the section 3.5.

3.1 EEGLAB's ICLabel

EEGLAB is a widely used open-source MATLAB toolbox for analysis of EEG data. It is an open-source software project of the Swartz Center for Computational Neuroscience (SCCN) of the University of California San Diego (UCSD) [61]. The data analysis functions available on EEGLAB include: data filtering, data epoch extraction, baseline removal, average reference conversion, data resampling and extraction of data epochs. EEGLAB also includes methods allowing users to remove data channels, epochs, and components dominated by non-neural artifacts, by accepting or rejecting visually-cued EEGLAB recommendations derived from signal processing and information measures [74]. EEGLAB uses a single structure to store data, acquisition parameters, events, channel locations, and epoch information as an EEGLAB dataset.

ICLabel is an EEGLAB plug-in presented by Pion-Tonachini and colleagues in 2019 [3]. It's a new CNN-based deep learning architecture to automatically classify ICs into seven classes i.e. brain, eye, heart, muscle ,line noise , channel noise and other. Moreover, they proposed to train and validate the classifier using an ICs dataset that was collected in a crowdsourcing way on a proper website [75].

- **Brain** ICs contain activity from locally synchronously activity in one (or sometimes two well-connected) cortical patches, they tend to have power spectral densities with

inversely related frequency and power, and exhibit increased power in frequency bands between 5 and 30 Hz.

- **Muscle** ICs, are surface EMG measures recorded using EEG electrodes, they contain activity originating from groups of muscle motor units (MU) and strong high-frequency broadband activity aggregating many MU action potentials (MUAP) during muscle contractions and periods of static tension.
- **Eye** ICs describe activity originating from the eyes such as horizontal eye movements, ICs blinks and vertical eye movements.
- **Heart** ICs are rare electrocardiographic (ECG) signals recorded using scalp EEG electrodes. They are recognizable by the clear QRS complexes.
- **Line noise** ICs capture the effects of line current noise emanating from nearby electrical fixtures or poorly grounded EEG amplifiers. They are recognizable by their high concentration of power at either 50 Hz or 60 Hz.
- **Channel noise** ICs indicate that some portion of the signal recorded at an electrode channel is already nearly statistically independent of those from other channels. These components can be produced by high impedance at the scalp-electrode junction or physical electrode movement, and are typically an indication of poor signal quality or large artifacts affecting single channels.
- **Other** ICs are ICs that fit none of the previous types.

ICLabel is trained by Pion-Tonachini and collegues with a dataset that has been drawn from 6352 EEG recordings collected from storage drives at the Swartz Center for Computational Neuroscience (SCCN) at UC San Diego. These recordings result from different studies where the participants were involved in different activities such as pressing bottoms or throwing darts. Also the numbers of electrodes and their positions differ across studies.

The website [75] has a key role in collect labels, from contributors. The website collected over 34,000 suggested labels on over 8000 ICs, each labeled IC has an average of 3.8 suggested labels associated with it. These labels are processed using the crowd labeling (CL) algorithm and the "crowd labeling latent Dirichlet allocation" (CL-LDA) to estimates "true labels" as a compositional vector (vector of nonnegative elements that sum to one) for each IC using the redundant labels from different labelers [75]. All these labels are used to trian the classifier. Figure 3.1 presents an example of output from ICLabel for a set of ICs. Each topographic map represents one IC with its weights for each electrode, and each subplot includes the suggested class ICLabel assigns to the topography. For example, IC 19 is assigned to **brain** with a probability of 99.5% and IC 29 is assigned to **line noise** with a probability of 93.6%.

Further details are provided to the user to check the time-course of the corresponding IC, its power spectrum, and the other classes probability, as shown in Figure 3.2. Particularly, the following information are provided:

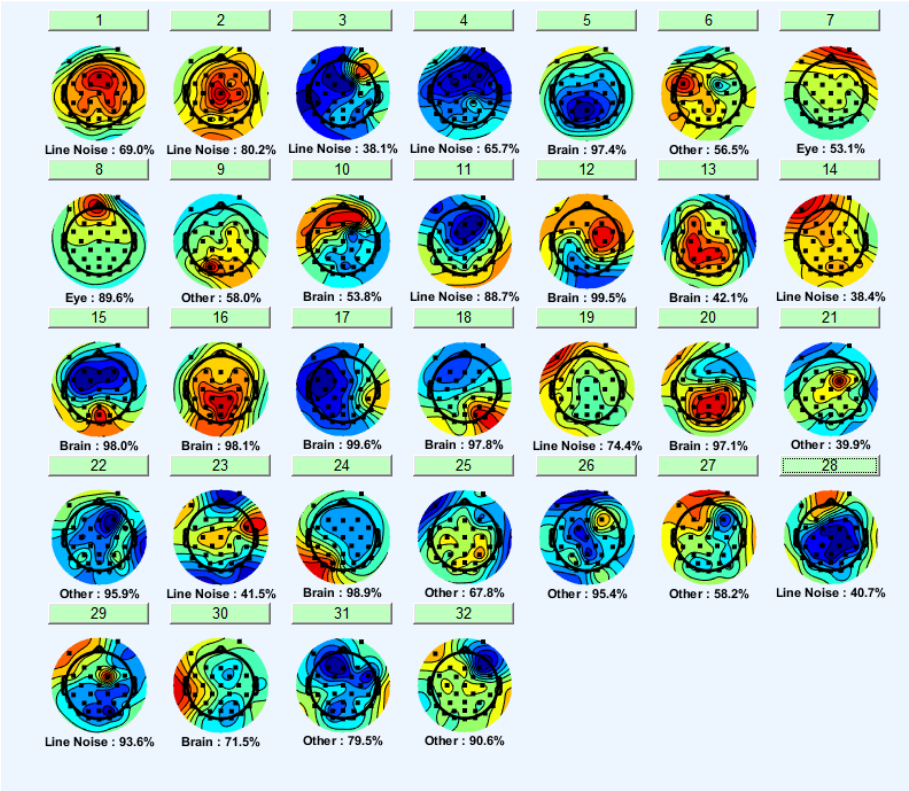

Figure 3.1: ICs ICLabel classification example

a. Scalp topography (top left panel in figure 3.2): it shows what weight the IC has on each electrode by interpolating and extrapolating IC projections to each electrode position into a standard projection image across the scalp. The green color code represents no effect, red and blue show positive and negative contributions, respectively.

b. ERP image (bottom left panel in figure 3.2): ERP stands for "event related potential", which is the repeatable response of the brain to a stimulus.

c. Component time series (top right panel in figure 3.2): it shows a segment of activity from entire time course of the IC.

d. Activity power spectrum (bottom right panel in figure 3.2): it shows the power spectrum of the IC activity across the entire dataset. It's calculated using a variation of Welch's method .

e. IC number and percent data variance accounted for (middle left panel in figure 3.2): Percent data variance accounted for describing how much of the original variance in the channel data can be attributed to this IC.

f. Probability results (middle top panel in figure 3.2): it shows the probability that the IC feature belongs to each class.

Pion-Tonachini and colleagues [3] has already compared ICLabel against other publicly available EEG IC classifiers as MARA, ADJUST, FASTER, SASICA and IC-MARC, that

are only a restricted group of automatic method classifier: publicly available, that do not require any information beyond the ICA-decomposed EEG recordings, with generally available meta-data such as electrode locations, and that have at minimum a category for Brain ICs. This comparison demonstrates that ICLabel computes labels ten times faster than those classifiers. Also recent application of ICLabel on other different classifiers, such as standardized and automated EEG processing open-source scripts (EPOS), demonstrates its good performances. One other example, comparing EPOS with the Harvard automated processing pipeline for EEG (HAPPE) [76], with the same dataset, as maid by Rodriugues et al. [77], ICLabel showed similar accuracy performances but more variance restricting and even less residual artifact prone fashion. This proves once again the optimal performances of ICLabel and its promising value to support experts and non-experts in objective distinguishing brain-related ICs from purely artifactual ICs.

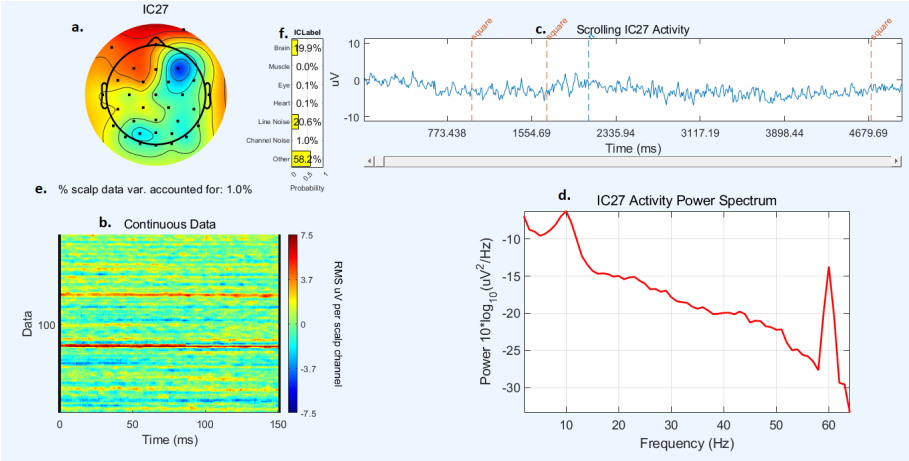

Figure 3.2: Example of ICLabel output

3.2 Short-Time Fourier Transform

The short-time Fourier Transform (STFT) is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time [78]. The STFT has a fundamental property that simplifies the interpretation of the resultant distribution, i. e. magnitude-wise shift invariance in both time and frequency. In practice, the procedure for computing STFTs is to divide a longer time signal into shorter segments of equal length and then compute the Fourier transform separately on each shorter segment. This reveals the Fourier spectrum on each shorter segment. One then usually plots the changing spectra as a function of time, known as a spectrogram [79]. The Fourier transform decomposes space-time dependent functions into functions depending on spatial frequency or temporal frequency.

The mathematical definition of continuos time short-time Fourier transform is:

$$\mathbf{STFT}\left\{\mathbf{x(t)}\right\}(\tau,\omega) \equiv \mathcal{X}(\tau,\omega) = \int_{-\infty}^{\infty} \mathbf{x(t)}\omega(\mathbf{t}-\tau)\mathbf{e}^{-\mathbf{i}\omega\mathbf{t}}\mathbf{dt} \tag{3.1}$$

where $\omega(\tau)$ is the window function, commonly a Hann window or Gaussian window centered around zero, and $\mathbf{x}(\mathbf{t})$ is the signal to be transformed $\mathcal{X}(\tau, \omega)$is essentially the Fourier transform of $\mathbf{x}(\mathbf{t})\omega(\mathbf{t} - \tau)$ a complex function representing the phase and magnitude of the signal over time and frequency.

3.3 Gradient-weighted Class Activation Mapping (Grad-CAM)

Grad-CAM is a specific CAM implementation that uses class-specific gradient information to localize important regions of the data. The algorithm identifies which EEG features were used by a network to classify an EEG epoch as favorable or unfavorable outcome, and also to understand failures of the network. A heatmap is created by the size of the features maps from the last convolutional layer of the classification network to highlight specific regions of the EEG segments which supported the classification as favorable outcome. Grad-CAM is also used to highlights the so-called "counterfactual explanations", i.e. regions that if removed could change the algorithm's classification. With this approach Grad-CAM can also highlights regions supportive of a classification as favorable outcome [80].

In order to obtain the class discriminative localization map Grad-CAM $\mathbf{L^c_{Grad-CAM}} \in \mathbb{R}^{\mathbf{c}\times\mathbf{t}}$ of width u and height v for any class c, compute the gradient of the score for class c, $\mathbf{y^c}$ (before the softmax), with respect to feature maps $\mathbf{A^k}$ of a convolutional layer, i.e. $\frac{\partial \mathbf{y^c}}{\partial \mathbf{A^k}}$. These gradients flowing back are global-average-pooled to obtain the neuron importance weight $\alpha^{\mathbf{c}}_{\mathbf{k}}$:

$$\alpha^{\mathbf{c}}_{\mathbf{k}} = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{global\ average\ pooling} \underbrace{\frac{\partial \mathbf{y^c}}{\partial \mathbf{A_{ij}^k}}}_{gradients\ via\ backprop} \tag{3.2}$$

This weight $\alpha^{\mathbf{c}}_{\mathbf{k}}$ represents a partial linearization of the deep network downstream from A, and captures the 'importance' of feature map k for a target class c. Then, a weighted combination of forward activation maps, and follow it by a ReLU, are applied to obtain:

$$\mathbf{L^c_{Grad-CAM}} = ReLU \underbrace{\left(\sum_k \alpha^c_k \mathbf{A^k}\right)}_{linear\ combination} \tag{3.3}$$

In general, $\mathbf{y^c}$ need to be the class score produced by an image classification CNN. It could be any differentiable activation including words from a caption or the answer to a question, as described by Selvaraju et al. [81]. In figure 3.3 there's an example and explanation of the Grad-CAM structure.

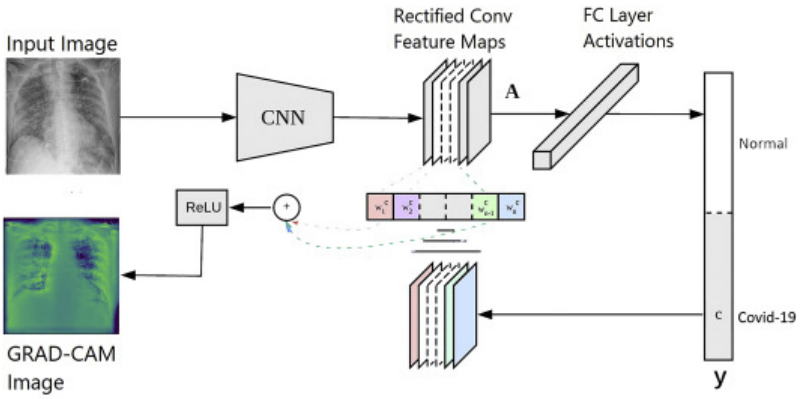

Figure 3.3: Example of Grad-CAM applied to highlights the damages of Covid-19 on lungs.

3.4 CNN

The CNN used in this thesis work is composed by two dimensional (2D) convolutional layers that were created with *convolutional2dLayer*. Between each convolutional layer there are Batch normalization layers, created with the function *batchNormalizationLayer*, that normalize the activations and gradients propagating through the network, making it the training an easier optimization problem. The batch normalization layer is followed by a nonlinear activation function, i.e. the rectified linear unit (ReLU), created with the function *reluLayer*. Convolutional layers (with activation functions) are followed by a down-sampling operation that reduces the spatial size of the feature map and removes redundant spatial information. Down-sampling makes it possible to increase the number of filters in deeper convolutional layers without increasing the required amount of computation per layer. For the down-sampling were created a max pooling layer, using *maxPooling2dLayer*. The max pooling layer returns the maximum values of rectangular regions of inputs, specified by the first argument, poolSize. The 'Stride' name-value pair argument specifies the step size that the training function takes as it scans along the input. The convolutional and down-sampling layers are followed by one fully connected layers, created with the function *fullyConnectedLayer* that connect the neurons to all the neurons in the preceding layer. This layer combines all the features learned by the previous layers across the image to identify the larger patterns. The last fully connected layer combines the features to classify the images. Therefore, the OutputSize parameter in the last fully connected layer is equal to the number of classes in the target data. Than, the softmax activation function, *softmaxLayer*, normalizes the output of the fully connected layer. The output of the softmax layer consists of positive numbers that sum to one. The final layer is the classification layer, *classificationLayer*. This layer uses the probabilities returned by the softmax activation function for each input, to assign the input to one of the mutually exclusive classes and compute the loss.

3.5 Implementation

Before creating the CNN as described before, five tools were devolped during this thesis work to adapt the data from the dataset to the input that the CNN requires to do the training of the net. The first tool is *formatDATA*(), this function takes the folder "trail" where are contained the time course for each ICs of several acquisitions of the EEG signal of the subject, it puts together all the ICs classified as brain and not brain in two folders "dataIC" and "dataNOIC" contained in the MATLAB struct "EEG". The names of the ICs already classified as brain or not brain are contained in the folders "brain_ic_vs" contained in the struct "comp_class". This classification was made by experts and professors at University of La Sapienza of Rome. Then the tools *estrazioneDATA*() used the function *formatDATA*() to extract the data for each subject saving each "EEG" MATLAB structure of the subject into the folders "EEG" and calculate the maximum and minimum of the amplitude of all the ICs. The tool *createImages*() uses this the maximum and minimum of the amplitude to calculate and normalize each ICs, the tool uses a short-time Fourier Transform (STFT) that is generally used to analyze how the frequency content of a nonstationary signal changes over time. The function *spectrogram*(), plot the spectrogram with a sample rate of 256 Hz (as used in the dataset), produced by the STFT, of all the ICs extracted before. The spectrograms have a range of frequency between 0 Hz and 120 Hz and a time renage of 0 to 6 minutes that correspond with the acquisition time of the original data. The spectrograms are created for each IC of each subject and saved in a folder named "BrainIC" and "BrainNOIC" respectively for the ICs classified as brain and those classified as not brain (each subject has is folders named "BrainIC" followed by the number of the subject).

The CNN developed during this thesis work is composed by three, two dimensional (2D) convolutional layers, two pooling layers, three batch normalization layers, three ReLU activation functions, one fully connected layer, one softmax layer and one classification layer. The three convolutional layers were created using *convolutional2dLayer* with a filter size of 5-by-5 and a number of layer, respectively, of 8 for the first convolutional layer, 16 for the second, and 32 for the third, these are the number of neurons in the layer that are connected to the same region of the input. Between each convolutional layer there are Batch normalization layers, created with the function *batchNormalizationLayer*. Convolutional layers (with activation functions) are followed by a max pooling layer, using *maxPooling2dLayer* with size of the rectangular region of [2,2]. The convolutional and down-sampling layers are followed by one fully connected layers, created with the function *fullyConnectedLayer* . The OutputSize parameter is 2, corresponding to the 2 classes of output: ICs classified as brain "BrainIC" and ICs classified as not brain "BrainNON_IC". Than, there is the softmax activation function, *softmaxLayer*, and at the end the classification layer, *classificationLayer*.

To train the network we had to choose some subjects because the number of samples differs from one subject to another and to have all the spectrograms with the same di-

mensions and do not eliminate too much samples from some subjects we decided to don't consider some subjects. The ICs used for the training were 2008, randomly selected in order to have 256 ICs for each of the two classes of output. By doing this, we used 512 ICs for the training that correspond to the 19% of the total. 215 ICs, that belongs to the two subjects excluded from training, that corresponds to the 8% of the total, were used for the test and the validation of the net. Respectively all the ICs of one subject 107 were used for the test and the 108 ICs of the other subject were used for the validation. With the data correctly elaborated in the two folders "BrainIC" and "BrainNON_IC", we proceeded with the training tests of the net with stochastic gradient descent algorithm with momentum (SGDM) [82] and ADAM. These two algorithms are considered the most efficient algorithms for stochastic optimization in the training process of convolutional neural network, as described by Kingma and colleagues [83]. SGDM is an algorithm that can oscillate along the path of steepest descent towards the optimum, adding a momentum term to the parameter update to reduce this oscillation. ADAM is another algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments [83]. It uses a parameter update with single learning rate, but with an added momentum term. It keeps an element-wise moving average of both the parameter gradients and their squared values.

After some tests changing the parameters of training we compared the performances between ICLabel and the best results of the CNN's training, as described in the chapter 4, to see how far the performances of the CNN are from those of ICLabel. We also applied GradCAM to the CNN to understand what are the most important parts of the spectrogram that the net uses to classify the data. The tool *applyGradCAM*(), simply applied GradCAM, with the MATLAB built-in function *gradCAM*(), to the CNN, giving as output an heatmap. This heatmap were overlaid on an IC of the dataset to higlights the regions of the spectrogram that the CNN uses to classy the IC into brain IC or not.

3.6 Dataset

The dataset used was created by Cosync Lab, that is a scientific laboratory of the Department of Psychology of University of La Sapienza of Rome. The dataset was prevously used by Milad in his MSc thesis (april, 2022) and is composed by 31 folders, one for each subjects, where are contained the ICA classification outputs. In each folder there is a MATLAB matrix ".mat" file containing:

- **options** with parameters used by the algorithm, E.g., filters, removed epochs, channel number, etc.
- **comp_class** that is a MATLAB structure with the classification results. In this structure are saved the following fields :
  - **trial** that is the time course for each component.

- **topo** that is a matrix with dimension of number of channels by number of ICs, and corresponds to the ICA spheres.
- **unmixing** that is a matrix with dimension of number of ICs by number of channels, and corresponds to the ICA weights.
- **topolabel** contains the channels labels
- **class** that is a MATLAB structure containing the classification results. In particular the field "brain_ic_vs" is a vector with the components classified as brain. It also contains the values for each threshold for each component (e.g., "elc_signal_correlation").

The dataset contains a total of 2677 ICs of which 2008 ICs, that correspond at the 75% of the total, were used to the training the CNN model (see Section 3.4). These ICs are subdivided into the two folders "BrainIC" and "BrainNON_IC" respectively, 256 in the folder "BrainIC" and 1499 in the folder "BrainNON_IC". For the validation o the CNN were used 108 ICs that corresponds to the 4% and for the test were used 107 ICs.

Chapter 4

Results and discussion

What we expected from the comparison between ICLabel and the CNN implemented is that the accuracy of ICLabel could be better than the CNN's one, because the training of ICLabel is based on more data coming from different acquisition tests, instead the CNN implemented is based on a single dataset with only 31 subjects. Also, ICLabel classifies the data in seven different class instead CNN classifies the data in only two classes, this could determine differences in accuracy results.

After several training tests of the CNN, the best result in accuracy was obtained with ADAM with an initial learning rate of 0.01, a maximum number of epochs of 20 and with a frequency of network validation in number of iterations of 50 as in figure 4.1. The CNN demonstrates good results for validation accuracy 98.15% and 99.07% for test accuracy. We might explain this performance considering that: firstly the CNN implemented is a simple net work that classifies only two classes i.e. it assigns the artifacts in only one class; secondly, the training of the CNN is based on only 256 ICs for Brain class and 256 ICs for not Brain class from the same dataset. We also made tests and validation on two different subjects from the same dataset and maybe the application of this net on other data could give less accurate results.

Comparing the results obtained by ICLabel and the simple CNN we can affirm that the number of ICs classified by the simple CNN of one subject are the same of those of ICLabel classification on the same subject. This is a good result, but we have to consider that this comparison is made with only one subject, test and validation consider only one subject, and the training uses not all the subjects. To do things properly we should have trained several times the net work choosing two different subjects each time for test and validation and using the others for training. However the results obtained are good and better of what we expected.

What we expected by the application of GradCAM on the CNN is closed to the results obtained. Certainly, what we expect is that GradCAM highlights some regions of the IC spectrogram that are important for the classification, and that this region could be related to certain specific frequency bands of the EEG. What we obtained is represented in figure 4.2, where the spectrogram of a brian IC represents the time-frequency visualization of

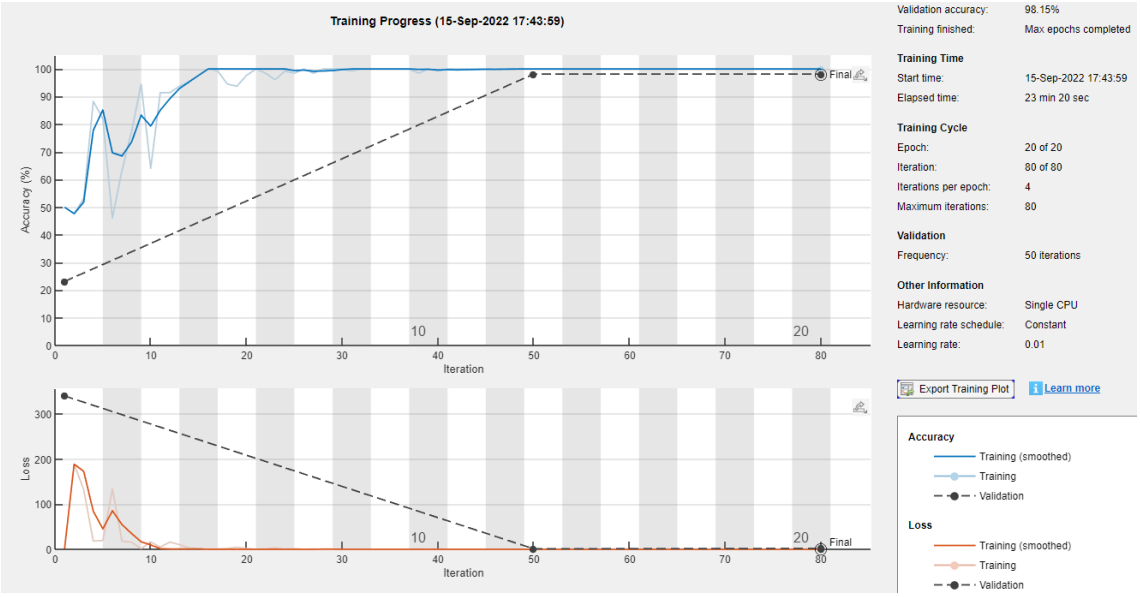

Figure 4.1: Best training results of the CNN implemented, with ADAM opitmization.

the IC (Fig. 4.2 a) and the GradCAM result (Fig. 4.2 b), on the same IC, represents the important regions for the classification, of the spectrogram. These results are what we expected.

In the figure 4.3 there's the rapresentation of the spectrogram of a not brain IC and the GradCAM result related to this IC. In this case we could notice that the regions highlighted are bigger then the figure 4.2 and not related to only one frequency. We could speculate that, as expected, the CNN is driven by components such as the power line noise (50 Hz and higher harmonics) to identify non-brain components, while it focuses on the range 1-30 Hz to identify brain components.

4.1 Conclusions

As described, our CNN has good performances, we compared them to those of ICLabel as in thesis work of Milad [5] where 99% of the ICs of the dataset that we also used are classified as brain. The CNN the validation accuracy is of 98.15% and the test accuracy is of 99% similar to the performances of ICLabel. We applied GradCAM to better understand how ICLabel and similar CNN classify the EEG artifacts. IClabel classifies the artifacts in seven different classes and is trained on several dataset with a huge number of ICs. Our CNN si trained in a small dataset that doesn't contain all the variability on which ICLabel is trained. So, more studies are needed. What could be done in the future is to create a CNN more similar to the IClabel's one with the same layers with seven output classes and three inputs. Maybe this networks could be trained in a dataset similar to the dataset used by Pion-Tonachini to trian ICLabel. Even though the results on GradCAM need more investigations, this work was useful to set up a general framework to explain the ICs automatic classification and to reveal the importance of the 50Hz line noise in the

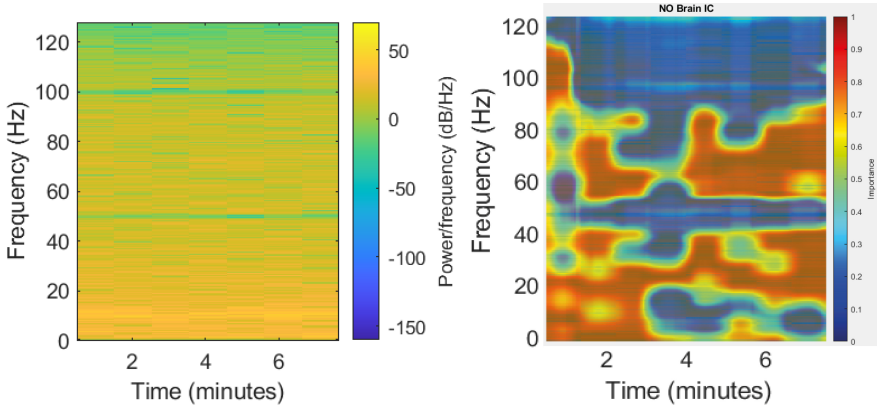

Figure 4.2: Brain IC spectrogram and GradCAM heatmap output.

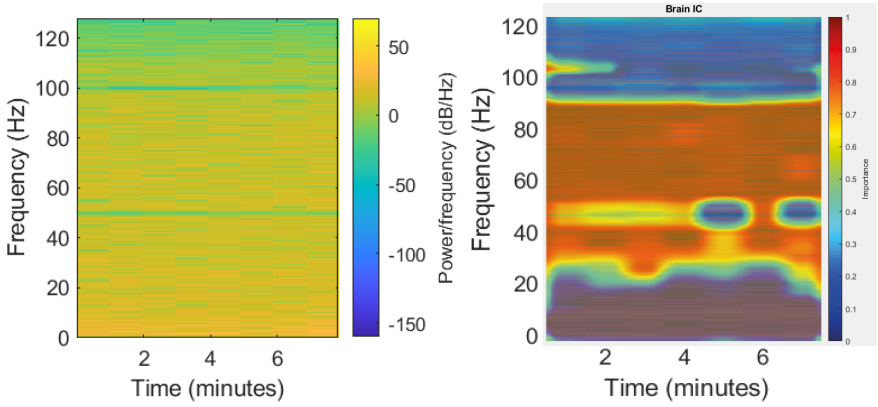

Figure 4.3: Not brain IC spectrogram and GradCAM heatmap output.

classification of the ICs.

Bibliography

[1] A. S. Al-Fahoum and A. A. Al-Fraihat, "Methods of eeg signal features extraction using linear analysis in frequency and time-frequency domains," *International Scholarly Research Notices*, vol. 2014, 2014.

[2] T. J. Sejnowski, "The unreasonable effectiveness of deep learning in artificial intelligence," *Proceedings of the National Academy of Sciences*, vol. 117, no. 48, pp. 30 033–30 038, 2020.

[3] L. Pion-Tonachini, K. Kreutz-Delgado, and S. Makeig, "Iclabel: An automated electroencephalographic independent component classifier, dataset, and website," *NeuroImage*, vol. 198, pp. 181–197, 2019.

[4] A. Tandle, N. Jog, P. D'cunha, and M. Chheta, "Classification of artefacts in eeg signal recordings and overview of removing techniques," *International Journal of Computer Applications*, vol. 975, p. 8887, 2015.

[5] N. N. Milad, "Advanced pipelines for artifact removal from eeg data," M.S. thesis, Dept. Eng. Inf., University of Padua, Italy, april 2022.

[6] B. Farnsworth, "What is eeg (electroencephalography) and how does it work?" *imotions. https://imotions. com/blog/what-is-eeg*, vol. 8, 2018.

[7] Wikipedia contributors, "10–20 system (eeg) — Wikipedia, the free encyclopedia," 2022, [Online; accessed 15-September-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=10%E2%80%9320_system_(EEG)&oldid=1076814653

[8] D. N. Dog. Dog not dog. Available [25/09/2019]: https://bit.ly/3iKUOho.

[9] B. Mijović, "Motion-related artifact removal using artifact subspace reconstruction," Aug 2021. [Online]. Available: https://medium.com/@boggisha/motion-related-artifact-removal-using-artifact-subspace-reconstruction-b622d416f30c

[10] C. S. Von Bartheld, J. Bahney, and S. Herculano-Houzel, "The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting," *Journal of Comparative Neurology*, vol. 524, no. 18, pp. 3865–3895, 2016.

[11] Wikipedia contributors, "Neuron — Wikipedia, the free encyclopedia," 2022, [Online; accessed 29-July-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Neuron&oldid=1101029060

[12] M. A. Lopez-Gordo, D. Sanchez-Morillo, and F. P. Valle, "Dry eeg electrodes," *Sensors*, vol. 14, no. 7, pp. 12 847–12 870, 2014.

[13] X. Jia and A. Kohn, "Gamma rhythms in the brain," *PLoS biology*, vol. 9, no. 4, p. e1001045, 2011.

[14] H. H. Jasper, "Recent advances in our understanding of ascending activities of the reticular system." 1958.

[15] G. E. Chatrian, E. Lettich, and P. L. Nelson, "Ten percent electrode system for topographic studies of spontaneous and evoked eeg activities," *American Journal of EEG technology*, vol. 25, no. 2, pp. 83–92, 1985.

[16] R. Oostenveld and P. Praamstra, "The five percent electrode system for high-resolution eeg and erp measurements," *Clinical neurophysiology*, vol. 112, no. 4, pp. 713–719, 2001.

[17] J. S. Kumar and P. Bhuvaneswari, "Analysis of electroencephalography (eeg) signals and its categorization–a study," *Procedia engineering*, vol. 38, pp. 2525–2536, 2012.

[18] W. Mumtaz, S. Rasheed, and A. Irfan, "Review of challenges associated with the eeg artifact removal methods," *Biomedical Signal Processing and Control*, vol. 68, p. 102741, 2021.

[19] K. S. Madhavi, M. Deeksha, S. Gayathri, N. U. N. Venkat, and H. K. Goru, "Denoising of ocular artifacts from single-channel eeg signals: A review," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1. IEEE, 2022, pp. 1002–1007.

[20] L. Sörnmo and P. Laguna, *Bioelectrical signal processing in cardiac and neurological applications*. Academic press, 2005, vol. 8.

[21] T. Opthof, "The normal range and determinants of the intrinsic heart rate in man," *Cardiovascular research*, vol. 45, no. 1, pp. 177–184, 2000.

[22] L. F. Azevedo, P. d. S. Perlingeiro, D. T. Hachul, I. L. Gomes-Santos, P. C. Brum, T. Allison, C. Negrão, and L. De Matos, "Sport modality affects bradycardia level and its mechanisms of control in professional athletes," *International journal of sports medicine*, vol. 35, no. 11, pp. 954–959, 2014.

[23] J. A. Urigüen and B. Garcia-Zapirain, "Eeg artifact removal—state-of-the-art and guidelines," *Journal of neural engineering*, vol. 12, no. 3, p. 031001, 2015.

[24] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[25] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[26] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021.

[27] S. A. R. Shah, H. Arshad, M. Farhan, S. S. Raza, M. M. Khan, S. Imtiaz, G. Shahzadi, M. A. Qurashi, and M. Waseem, "Sustainable brick masonry bond design and analysis: An application of a decision-making technique," *Applied Sciences*, vol. 9, no. 20, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/20/4313

[28] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.

[29] A. Zancanaro, "Machine and deep learning algorithms for the analysis of eeg signals for neuroscience and motor rehabilitation," M.S. thesis, Dept. Eng. Inf., Università degli Studi di Padova, Italy, July 2021.

[30] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai—explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, p. eaay7120, 2019.

[31] D. T. Barus, F. Masri, and A. Rizal, "Ngboost interpretation using lime for alcoholic eeg signal based on gldm feature extraction," in *Proceedings of the Computational Methods in Systems and Software*. Springer, 2020, pp. 894–904.

[32] S. M. Lundberg and S.-I. Lee, "Consistent feature attribution for tree ensembles," *arXiv preprint arXiv:1706.06060*, 2017.

[33] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention-based encoder-decoder networks," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1875–1886, 2015.

[34] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[35] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[36] C. A. Ellis, R. L. Miller, and V. D. Calhoun, "A systematic approach for explaining time and frequency features extracted by convolutional neural networks from raw electroencephalography data," *Frontiers in Neuroinformatics*, vol. 16, 2022.

[37] C. Rashmi and C. Shantala, "Eeg artifacts detection and removal techniques for brain computer interface applications: a systematic review," *International Journal of Advanced Technology and Engineering Exploration*, vol. 9, no. 88, p. 354, 2022.

[38] D. Gorjan, K. Gramann, K. De Pauw, and U. Marusic, "Removal of movement-induced eeg artifacts: current state of the art and guidelines," *Journal of neural engineering*, 2022.

[39] M. K. Islam, A. Rastegarnia, and Z. Yang, "Methods for artifact detection and removal from scalp eeg: A review," *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 46, no. 4-5, pp. 287–305, 2016.

[40] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, "Emg and eog artifacts in brain computer interface systems: A survey," *Clinical neurophysiology*, vol. 118, no. 3, pp. 480–494, 2007.

[41] J.-M. Bollon, R. Chavarriaga, J. d. R. Millán, and P. Bessiere, "Eeg error-related potentials detection with a bayesian filter," in *2009 4th International IEEE/EMBS Conference on Neural Engineering*. IEEE, 2009, pp. 702–705.

[42] J. M. Antelis, L. Montesano, A. Ramos-Murguialday, N. Birbaumer, and J. Minguez, "On the usage of linear regression models to reconstruct limb kinematics from low frequency eeg signals," *PloS one*, vol. 8, no. 4, p. e61976, 2013.

[43] V. Krishnaveni, S. Jayaraman, L. Anitha, and K. Ramadoss, "Removal of ocular artifacts from eeg using adaptive thresholding of wavelet coefficients," *Journal of neural engineering*, vol. 3, no. 4, p. 338, 2006.

[44] J. V. Stone, "Independent component analysis: a tutorial introduction," 2004.

[45] A. K. Abdullah, C. Z. Zhang, A. A. A. Abdullah, and S. Lian, "Automatic extraction system for common artifacts in eeg signals based on evolutionary stone's bss algorithm," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[46] A. Vergult, W. De Clercq, A. Palmini, B. Vanrumste, P. Dupont, S. Van Huffel, and W. Van Paesschen, "Improving the interpretation of ictal scalp eeg: Bss–cca algorithm for muscle artifact removal," *Epilepsia*, vol. 48, no. 5, pp. 950–958, 2007.

[47] L. Cao, K. S. Chua, W. Chong, H. Lee, and Q. Gu, "A comparison of pca, kpca and ica for dimensionality reduction in support vector machine," *Neurocomputing*, vol. 55, no. 1-2, pp. 321–336, 2003.

[48] X. Yong, R. K. Ward, and G. E. Birch, "Artifact removal in eeg using morphological component analysis," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 345–348.

[49] A. Tharwat, "Independent component analysis: An introduction," *Applied Computing and Informatics*, 2020.

[50] G. R. Naik and D. K. Kumar, "An overview of independent component analysis and its applications," *Informatica*, vol. 35, no. 1, 2011.

[51] M. K. I. Molla, M. R. Islam, T. Tanaka, and T. M. Rutkowski, "Artifact suppression from eeg signals using data adaptive time domain filtering," *Neurocomputing*, vol. 97, pp. 297–308, 2012.

[52] M. A. Uusitalo and R. J. Ilmoniemi, "Signal-space projection method for separating meg or eeg into components," *Medical and biological engineering and computing*, vol. 35, no. 2, pp. 135–140, 1997.

[53] C.-Y. Chang, S.-H. Hsu, L. Pion-Tonachini, and T.-P. Jung, "Evaluation of artifact subspace reconstruction for automatic eeg artifact removal," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 1242–1245.

[54] ——, "Evaluation of artifact subspace reconstruction for automatic artifact components removal in multi-channel eeg recordings," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 4, pp. 1114–1121, 2019.

[55] X. Chen, Q. Liu, W. Tao, L. Li, S. Lee, A. Liu, Q. Chen, J. Cheng, M. J. McKeown, and Z. J. Wang, "Remae: User-friendly toolbox for removing muscle artifacts from eeg," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2105–2119, 2020.

[56] R. J. Kobler, A. I. Sburlea, V. Mondini, and G. R. Müller-Putz, "Hear to remove pops and drifts: the high-variance electrode artifact removal (hear) algorithm," in *2019 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC)*. IEEE, 2019, pp. 5150–5155.

[57] L. Frølich, T. S. Andersen, and M. Mørup, "Classification of independent components of eeg into multiple artifact classes," *Psychophysiology*, vol. 52, no. 1, pp. 32–45, 2015.

[58] M. Chaumon, D. V. Bishop, and N. A. Busch, "A practical guide to the selection of independent components of the electroencephalogram for artifact correction," *Journal of neuroscience methods*, vol. 250, pp. 47–63, 2015.

[59] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "MNE software for processing MEG and EEG data," *NeuroImage*, vol. 86, pp. 446–460, 2014.

[60] I. Daly, R. Scherer, M. Billinger, and G. Müller-Putz, "Force: Fully online and automated artifact removal for brain-computer interfacing," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 23, no. 5, pp. 725–736, 2014.

[61] C. Brunner, A. Delorme, and S. Makeig, "Eeglab–an open source matlab toolbox for electrophysiological research," *Biomedical Engineering/Biomedizinische Technik*, vol. 58, no. SI-1-Track-G, p. 000010151520134182, 2013.

[62] A. Mognon, J. Jovicich, L. Bruzzone, and M. Buiatti, "Adjust: An automatic eeg artifact detector based on the joint use of spatial and temporal features," *Psychophysiology*, vol. 48, no. 2, pp. 229–240, 2011.

[63] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen, "Fieldtrip: open source software for advanced analysis of meg, eeg, and invasive electrophysiological data," *Computational intelligence and neuroscience*, vol. 2011, 2011.

[64] I. Winkler, S. Haufe, and M. Tangermann, "Automatic classification of artifactual ica-components for artifact removal in eeg signals," *Behavioral and brain functions*, vol. 7, no. 1, pp. 1–15, 2011.

[65] H. Nolan, R. Whelan, and R. B. Reilly, "Faster: fully automated statistical thresholding for eeg artifact rejection," *Journal of neuroscience methods*, vol. 192, no. 1, pp. 152–162, 2010.

[66] F. C. Viola, J. Thorne, B. Edmonds, T. Schneider, T. Eichele, and S. Debener, "Semi-automatic identification of independent components representing eeg artifact," *Clinical Neurophysiology*, vol. 120, no. 5, pp. 868–877, 2009.

[67] N. Nicolaou and S. J. Nasuto, "Automatic artefact removal from event-related potentials via clustering," *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 48, no. 1, pp. 173–183, 2007.

[68] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial eeg classification," *Journal of neuroscience methods*, vol. 274, pp. 141–145, 2016.

[69] Y. Li, H. Yang, J. Li, D. Chen, and M. Du, "Eeg-based intention recognition with deep recurrent-convolution neural network: Performance and channel selection by grad-cam," *Neurocomputing*, vol. 415, pp. 225–233, 2020.

[70] H. Ye, F. Gao, Y. Yin, D. Guo, P. Zhao, Y. Lu, X. Wang, J. Bai, K. Cao, Q. Song *et al.*, "Precise diagnosis of intracranial hemorrhage and subtypes using a three-dimensional joint convolutional and recurrent neural network," *European radiology*, vol. 29, no. 11, pp. 6191–6201, 2019.

[71] H. Chen, Y. Song, and X. Li, "Use of deep learning to detect personalized spatial-frequency abnormalities in eegs of children with adhd," *Journal of neural engineering*, vol. 16, no. 6, p. 066046, 2019.

[72] S. Jonas, A. O. Rossetti, M. Oddo, S. Jenni, P. Favaro, and F. Zubler, "Eeg-based outcome prediction after cardiac arrest with convolutional neural networks: Performance and visualization of discriminative features," *Human brain mapping*, vol. 40, no. 16, pp. 4606–4617, 2019.

[73] A. Arias-Duart, F. Parés, and D. Garcia-Gasulla, "Who explains the explanation? quantitatively assessing feature attribution methods," *arXiv preprint arXiv:2109.15035*, 2021.

[74] A. Delorme and S. Makeig, "Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis," *Journal of neuroscience methods*, vol. 134, no. 1, pp. 9–21, 2004.

[75] Iclabel website. Available: https://labeling.ucsd.edu/tutorial.

[76] L. J. Gabard-Durnam, A. S. Mendez Leal, C. L. Wilkinson, and A. R. Levin, "The harvard automated processing pipeline for electroencephalography (happe): standardized processing software for developmental and high-artifact data," *Frontiers in neuroscience*, vol. 12, p. 97, 2018.

[77] J. Rodrigues, M. Weiß, J. Hewig, and J. J. Allen, "Epos: Eeg processing open-source scripts," *Frontiers in neuroscience*, vol. 15, p. 663, 2021.

[78] E. Sejdić, I. Djurović, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," *Digital signal processing*, vol. 19, no. 1, pp. 153–183, 2009.

[79] Wikipedia contributors, "Short-time fourier transform — Wikipedia, the free encyclopedia," 2022, [Online; accessed 11-September-2022]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Short-time_Fourier_transform&oldid=1104785770

[80] S. Jonas, A. O. Rossetti, M. Oddo, S. Jenni, P. Favaro, and F. Zubler, "Eeg-based outcome prediction after cardiac arrest with convolutional neural networks: Performance and visualization of discriminative features," *Human brain mapping*, vol. 40, no. 16, pp. 4606–4617, July 2019.

[81] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[82] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*. PMLR, 2013, pp. 1139–1147.

[83] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.