Incremental learning and discrimination of imagined speech for EEG-based BCIs

by:

M.Sc. Jesús Salvador García Salinas

A Dissertation Submitted to the Program in Computer Science, Computer Science Department in partial fulfillment of the requirements for the degree of:

Doctor in Computer Science

at the:

Instituto Nacional de Astrofísica, Óptica y Electrónica
April, 2022
Tonantzintla, Puebla

Advisor:

Ph.D. Luis Villaseñor Pineda

Contents

List of Figures

List of Tables

Aknowledgments

The present work was partially supported by CONACyT (scholarship 401887).

The authors also thank CONACyT for the computer resources provided through the INAOE Supercomputing Laboratory’s Deep Learning Platform for Language Technologies.

Chapter 1

Introduction

Brain-Computer Interfaces (BCIs) are systems able to transform the brain signals into commands to control a device. Different instruments are used to acquire brain signals, in this study electroencephalography (EEG) was employed to record the brain electrophysiological signals. In particular, the use of EEG is of great interest due to its simple operation and low cost.

Former EEG-based BCIs used an external stimulus in which the brain activity related to such stimulus is known [Farwell and Donchin, 1988]. The present study is based on the use of an internal stimulus related to language, known as imagined speech, which is the action of imagining the diction of a word without emitting nor articulating any sound [Torres-García et al., 2016]. The use of imagined speech may provide a new communication channel for computers. Beyond aiding people with disabilities, this approach can change their interactions with electronic devices.

An imagined speech based BCI requires the processing of brain signals to detect brain activity related to a specific word. Previous EEG-based BCIs have proposed methods for feature extraction and analysis. Some have attempted to represent EEGs as multidimensional data and have analyzed these representations in different ways [Ji et al., 2016, Zhao et al., 2009, Li and Zhang, 2010, Lee et al., 2007]. This study proposed a new method that integrates the dimensions of an EEG signal to find appropriate patterns for imagined speech discrimination. Therefore, a multiple variable (multivariate) representation will be proposed, as well as a deep learning architecture. These methods will be compared in subsequent experiments.

The objective is to add new words to a previously generated imagined speech discrimination model. When a BCI is trained for a specific task, its extension requires retraining it by adding information from a new task [Lotte and Guan, 2010]. The proposal is that a network architecture in conjunction with transfer learning can extend an imagined speech model to recognize new imagined words. This can be considered an intra-subject transfer learning task. BCI transfer is commonly focused on inter-subject variability, i.e., extending a generated model to new subjects [Waytowich et al., 2016, Panagopoulos, 2017, Wei et al., 2018, He and Wu, 2017].

Previous studies on EEG-based BCIs have employed stimuli that activate or deactivate specific brain regions. This allows for oriented feature extraction in such regions, thereby facilitating decoding. For imagined speech, the language regions to be monitored have not been well established. Therefore, widespread brain areas were measured. Another consideration that must be taken is that BCI models must be developed individually for each subject due to the variability in brain connectivity.

Despite these challenges, imagined speech provides multiple advantages, such as the possibility of creating a large vocabulary of commands whose interpretation is natural for the BCI final user, compared with other approaches, such as motor imagery.

For imagined speech, as in other approaches, a data acquisition process is required. This can be a lengthy and stressful procedure, and several studies aimed to reduce the time and effort required.

The aim of this study was to increase the imagined speech vocabulary of a model to ease the data acquisition process. For this purpose, the following considerations were taken.

- Which feature extraction methods are more suitable to decode imagined speech?
- How to implement a model which allows to decode imagined speech?
- From an imagined speech trained model, how it could be adapted to increase the vocabulary?

1.1 Motivation

BCI models require a long procedure to acquire data. These models are specifically designed for each subject owing to the neural connection variability. To solve this issue, previous studies aimed to transfer models between different subjects. However, the emergence of new BCI approaches, such as imagined speech, introduces new challenges; in this case, the possibility of extending a model to include new imagined words for a single subject. This incremental approach is common in tasks such as image classification; however, its implementation in imagined speech is challenging due to the complexity of EEG signals and data limitations.

1.2 Problem statement

Brain activity can be obtained through different feature extraction methods according to the application domain. In addition, feature extraction usually depends on the cognitive tasks analyzed. Previous studies have addressed different cognitive tasks such as imagined movement. These studies take advantage of the fact that imagined movement is spatially related to specific brain areas, depending on limb movement. However, previous studies concluded that language activates multiple brain areas [McGuire et al., 1996, Shergill et al., 2003, Kober et al., 2001].

The BCIs have inherent issues that must be considered. In the first instance, the datasets were obtained using a different number of channels, sample rates and devices. In addition, the acquisition protocols are extremely different, there is no standard for imagined speech acquisition protocols, and each study has different vocabularies, number of subjects, acquisition time, number of epochs, number of word repetitions, etc.

Despite the previous limitations, the imagined speech has advantages over other BCI paradigms, the most relevant to this work is the implementation of a large vocabulary that could be increased over time by adding new words by incremental learning.

Two common problems in incremental learning are the *catastrophic forgetting*, which is a drastic decrease in the performance over the old classes, and *intransigence* which inhibits adaptation of a new class [Chaudhry et al., 2018].

1.2.1 Contributions

To the best of our knowledge, incremental learning of imagined speech tasks for BCIs has not been explored. Transfer learning approaches in BCIs are often oriented toward subject-to-subject transfer, i.e, inter-subject transfer [Waytowich et al., 2016, Panagopoulos, 2017, Wei et al., 2018, He and Wu, 2017]. Imagined speech introduces an intuitive way to apply a transfer that allows the extension of a BCI to recognize new imagined words.

The main contribution is the development of a model that allows the incremental learning of imagined speech. To achieve this, various neural network architectures and a novel loss function have been proposed.

The neural network architecture is employed for feature extraction and creates a set of centroids to represent classes. These centroids were employed for classification purposes; therefore, the distance between the input data and centroids was the primary parameter used to train the network. Moreover, the proposed architecture grows to allow the inclusion of new classes. When the data of new classes are added, new modules of the network are implemented and trained by considering the information of the original data.

1.2.2 Scope and limitations

Despite the capability of multivariate decompositions to reconstruct the EEG signals and find sources of activity [De Vos et al., 2007, Deburchgraeve et al., 2009, Weis et al., 2009], this work will not analyze the activation areas of the brain related to imagined speech.

Interpretation of neural networks is not aimed in this work.

1.3 Hypothesis

By identifying patterns of brain activation, it will be possible to represent imagined words and determine the degree of similarity between them. Characterization of brain activation will allow imagined speech discrimination and extension of the vocabulary of imagined words without loss of discrimination performance.

1.4 Research questions

- How a set of features can be extracted from EEG signals of imagined speech, allowing the discrimination of imagined words?
- Which methods are more suitable to add a new imagined word to an existing model?
- Which variables have to be considered to increase a model for different datasets?

1.5 Objective

To develop and evaluate a transfer learning model for imagined speech EEG to increase the vocabulary without significantly decreasing the recognition rates of the model. Moreover, this model will discriminate among imagined words with a similar recognition rate performance compared with related works.

1.6 Specific objectives

- To analyze methods for multivariate analysis of the EEG signals.
- To compare the state of the art methods to find a model which allows incremental learning.
- To design and evaluate a framework to accelerate the integration of new imagined words, considering the minimal decrements of the recognition rates.

Chapter 2

Theoretical Framework

In this Chapter, the background concepts used in this study are reviewed. In Section 2.1, the mathematical notation is presented. In Section 2.2, Parallel Factor Analysis (PARAFAC) is formally explained.

Section 2.3 introduces a dictionary representation algorithm. Moreover, the Bag of Features method is reviewed. A dictionary representation allows to obtain descriptive units from a set of features and then represent any signal using these units. Section 2.4 explains the architecture of the proposed neural networks. Finally, in Section 2.5, a definition of transfer learning is presented. In addition, different approaches that are commonly used in EEG transfer learning are explained.

2.1 Tensor algebra

The mathematical notations in this study are based on the definitions presented in [Lu et al., 2013, Cichocki et al., 2015]. The primary notations are listed in Table 2.1.

Table 2.1: Basic tensor notations

Symbol	Description
$\mathscr{A}$	Tensor
$\mathbf{A}$	Matrix
$\mathbf{a}$	Vector
a	Scalar
$\mathbf{A}_{(n)}$	Mode-n unfolding of tensor $\mathscr{A}$
$\mathbf{U}^{(n)}$	Mode-n projection matrix
$\mathbf{A}(i_1, i_2)$	Entry at the $i_1 th$ row and $i_2 th$ column of $\mathbf{A}$
$\mathbf{a} \circ \mathbf{b}$	Outer (tensor) product

2.2 Parallel Factor Analysis

In mathematics, multidimensional arrays are referred to as tensors. In physics, it generally refers to a tensor field, a generalization of vector fields, which is an association of a tensor with each point of a geometric space. In the following, a tensor can be considered as a generalization of vectors and matrices [Lu et al., 2013].

A tensor is a multidimensional or an N-way array, where N is the tensor order. The order N of a tensor is the number of dimensions and is also referred to as ways or modes [Devarajan, 2011]. An example of tensor $\mathscr{X}$ is shown in Fig. 2.1, where the modes are time, channels and frequency, and e represents the epoch number.

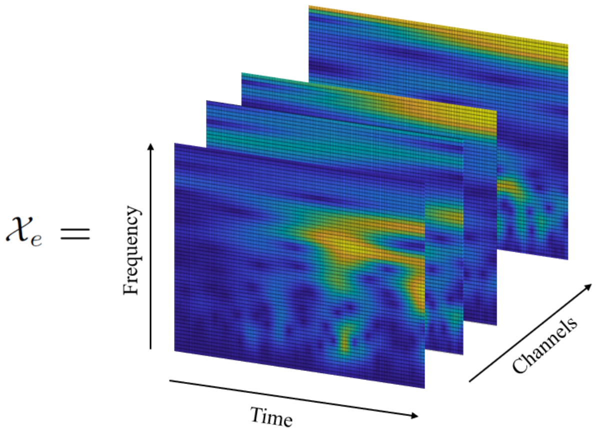


Figure 2.1: Tensor example.

Multilinear algebra introduces methods for tensor decomposition analysis. One of these, known as Parallel Factor Analysis (PARAFAC) or Canonical Decomposition (CP), was first proposed by [Hitchcock, 1927]. This method is based on Alternated Least Squares (ALS) of the different tensor projections.

Let $\mathscr{X}$ be an N-dimensional tensor, PARAFAC decomposes the tensor, as shown in Eq. 2.1.

$$\mathscr{X} = \sum_{p=1}^{P} u_p^{(1)} \circ u_p^{(2)} \circ \cdots \circ u_p^{(N)} \tag{2.1}$$

Where $\mathbf{U}^{(n)} = [u_1^{(n)}, u_2^{(n)}, ..., u_P^{(n)}]$ is a matrix denoting the components of the mode n. Operator $\circ$ is the n-mode outer product. Moreover, P are the number of extracted components in each mode and N is the number of modes in the tensor. Thus, the total number of components is given as $P \times N$.

The decomposition process is illustrated in Fig. 2.2, in which a three dimensional tensor $\mathscr{X}$ is decomposed into P factors. A three-dimensional tensor was chosen as a visual example. Nevertheless, this decomposition can be performed for any N-dimensional tensor.

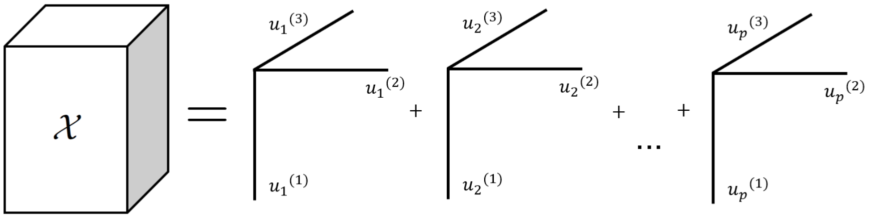


Figure 2.2: Tensor decomposition into P factors [Lu et al., 2013].

2.3 Dictionary generation

A dictionary is a set of elementary signals, also known as atoms, used to decompose a signal. A dictionary can be expressed as $D = [d_1, d_2, \cdots, d_j] \in \mathbb{R}^{j\times k}$, where j is the number of elements in the dictionary and k is the length of the elements. Subsequently, the signal Y is expressed by Eq. 2.2.

$$Y = Dx \tag{2.2}$$

Where Y denotes the reconstructed signal, D denotes the set of atoms (Dictionary) and x the set of coefficients.

Dictionary representation is usually restricted to finding a good signal reconstruction using as few coefficients as possible in x, which is known as sparsity. Dictionaries do not provide unique solutions and sparsity constraints are applied to determine a unique and compressed set of coefficients [Aharon et al., 2006, Tosic and Frossard, 2011]. This can be expressed by Eq. 2.3

$$\min_{x\in\mathbb{R}^k} ||Y - Dx||_2^2 + \lambda_1 ||x||_1 \tag{2.3}$$

Where λ_1 represents a Lagrange multiplier that restricts x to the L1 norm [Mairal et al., 2008].

2.3.1 Bag of features (BoF)

This method is based on *Vector Quantization* [Aharon et al., 2006, Zhuolin Jiang et al., 2013], which aims to achieve an automatic signal characterization by discretizing its representation [García-Salinas et al., 2017]. This is a specific case of dictionary generation in which the number of coefficients used for signal reconstruction is one. In signal analysis, many variations and adaptations of this method have been developed [Ameri et al., 2015, Ameri et al., 2016, Barthélemy et al., 2013, Gu et al., 2014, Pressel Coretto et al., 2017], that receive different names according to the application area. It is referred to as a bag of words in document classification, bag of instances in multiple instance learning, bag of frames in audio and speech recognition, bag of patterns in signal processing and pattern recognition, and bag of images in computer vision [Baydogan et al., 2013].

The BoF model was originally developed for document representations. The basic idea is to generate a dictionary called *codebook*, and the elements of this codebook are called *codewords*. Finally, histograms of the codewords are generated. Although the information order of words is ignored, the bag of words model has shown to be effective in capturing document information [Wang et al., 2013a]. The codewords are obtained from the codebook, which is commonly generated by a clustering procedure over segments of the analyzed object. Subsequently, each extracted segment is associated with a codeword, then it is possible to generate histograms to analyze the data.

A formalization of bag of features from [Gui and Yeh, 2014] is as follows. A time series is defined by the vector $x = (x_1, x_2, ..., x_i)$ for i-ordered samples. Each instance x^i is associated with the class $y \in \{1, 2, ..., C\}$, where C denotes the number of classes. To extract local patterns, a sliding window w over the time series is required. The displacement m of the window should not exceed its length $m \leq w$. The extracted sub-sequences will be $\lceil \frac{i-w+1}{m} \rceil$ and the dataset will have $n(\lceil \frac{i-w+1}{m} \rceil)$ sub-sequences. Subsequently, a clustering method is applied, and the centroids become the *codewords* that constitute a *codebook* $K \in \mathbb{R}^{(w \times d)}$, where K denotes a set of centroids.

This method assumes that a set of objects consists of finite features and that such objects are unordered sets of features. Moreover, histogram-based approaches ignore temporal ordering and therefore may not identify specific contents or patterns related to temporal information.

2.4 Neural networks

According to [Lecun et al., 2015], deep learning is a set of representation-learning methods with multiple levels of representation obtained by composing simple but non-linear modules. In addition, [Schmidhuber, 2015] established that neural networks consist of many simple, connected processors called neurons, which are activated by perceiving the environment, while others are activated through weighted connections from previous neurons.

Neural networks are based on a structure called a neuron, which is a linear function that takes a set of inputs X. These inputs are transformed by a set of weights W to generate a linear function that approximates input data [Haykin et al., 2009]. Moreover, a bias value b is added to each element of W, which is the constant term in a linear function. A formal representation of a neuron is as follows:

$$z = wx + b$$

Intuitively, the weights w are automatically learned using an optimization algorithm. Then, to predict the test instances, the dot product of the weights and input vectors is applied. These operations can be expressed as follows:

$$z = \big(\sum_{i=1}^{n} w_i x_i\big) + b$$

Fig. 2.3 shows the architecture of a neuron. From this perspective, the input vectors are obtained by the network, multiplied by the weights and finally summed, which essentially corresponds to the dot product with the bias added.

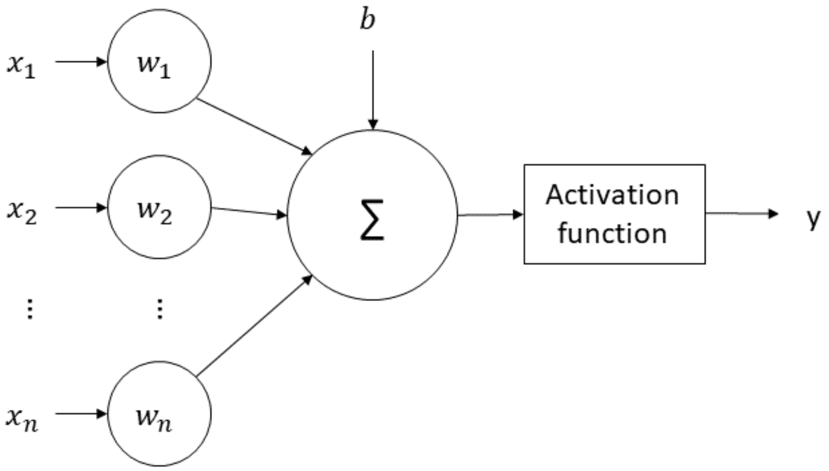


Figure 2.3: Graphical representation of a neuron. The input vector x is processed. The neuron uses the weights in w to conceptually perform the dot product of w and x, then add the bias value.

Once the input vectors, weights and bias are processed, an activation function is applied to z. Usually, the activation function $a = \sigma(z)$ maps the neuron information in z to a value between -1 and 1 in the case of the hyperbolic tangent function *tanh*, or 0 and 1 in the case of the *sigmoid* function. The sigmoid function is commonly used for prediction layers. For example, assuming a binary classification problem, it is expected that the network output determines whether the input corresponds to a specific class, i.e. a prediction close to one or close to zero. Therefore, the sigmoid function is convenient and typically implemented in the last layer of a network. The sigmoid activation function is defined as $\sigma(z) = \frac{1}{1+e^{-z}}$, which ensures a positive soft value for the prediction.

During the first stage of the learning phase, a crucial aspect is to observe the model performance to measure how well the parameters w and b fit the input data. To compute this fit, a loss function is required. Intuitively, the loss function is the primary method for measuring the error of the model at each training epoch. A common implementation based on logarithms is defined as follows:

$$L(a,y) = -(y \log a + (1-y)\log(1-a))$$

Where a is the predicted value, and it is compared with the true value y. It is expected that the computed value a approximates the real true value y. In this case, the loss function is applied individually to each input instance. Moreover, it is necessary to measure the behavior of the parameters over the entire sample set, for which a cost function is defined as:

$$C(W,b) = \frac{1}{m}\sum_{i=1}^{n} L(a^{(i)}, y^{(i)})$$

Where n denotes the training samples in the dataset. The objective is to minimize $C(W,b)$ to obtain the lowest cost value. In order to minimize the cost function, the W and b values must be updated using the gradient-descent approach. The computation of the partial derivatives to update the parameters W and b is called backpropagation, and extends to each neuron and layer of the network. The main equation is:

$$\frac{dC}{w_{ij}^{k}} = \frac{dC}{a_j^k}\frac{a_j^k}{w_{ij}^{k}} \tag{2.4}$$

At this point, another variable is required: learning rate, which is typically assigned to α. This value indicates the magnitude of the parameter update. The following equations were used to update the values:

$$w_{ij}^{k} = w_{ij}^{k} - \alpha\frac{dC(W,b)}{dw_{ij}^{k}} \tag{2.5}$$

$$b = b - \alpha \frac{dC(W,b)}{db} \tag{2.6}$$

These equations follow a gradient descent approach, in which each update to the parameters makes $C(W,b)$ closer to a minimum cost. If the minimum cost is reached, also $L(a,y)$ is reduced, therefore, parameters W and b fit the input data correctly.

2.4.1 Convolutional Neural Networks

To increase the feature extraction capabilities of neural networks, the addition of convolutional layers in the initial phase has been proposed. This has attracted considerable interest, particularly in image analysis. These convolution layers aim to find patterns across the input data by applying convolutional filters, which implies that a feature extraction process is added before neural network training.

The convolution of two functions $f(x)$ and $g(x)$ is defined in [Bracewell, 2000] as a continuous function:

$$f(x) * g(x) = \int_{-\inf}^{\inf} f(u)g(x-u)du$$

The discrete counterpart of such function is defined as:

$$f[m] + g[m] = \sum_{n} f[n]g[m-n]$$

In a practical scenario, $f(x)$ is the input data and $g(x)$ is the filter or kernel. The setting of the filters results in different feature extractions, a typical example is edge detection in images. There are well known filters for specific tasks, such as noise reduction, edge detection, sharpening, and blurring. Nevertheless, random filters are commonly used in neural networks when input data features are unknown.

An inherent issue of convolutions is that the border data is not treated equally as the inner data. Assume a vector X with ten elements, over which a convolution with a size of three kernel is applied. The resulting vector X' is reduced in size by $(X_n - K) + 1$, where X_n is the input size and K is the kernel size. It can be inferred that the edge data lose relevance in the resulting vector. A practical solution to this problem is to implement padding on the input. Padding involves the addition of data to the input edges to allow for complete convolution. Nevertheless, determining the value to be used for padding depends on the analyzed data. The most common approach is to use zeros; however, any other value is viable, e.g., ones, duplication of the edge value, or the mean of the next k values. By adding padding, the output size can be calculated as:

$$(X_n - K + 2P) + 1$$

Where P denotes the padding size. Note that the convolution kernel moves in steps of one across the input, but there are no restrictions on the step size, this displacement is called stride. As expected, it changes the output size of the data as follows:

$$\lfloor (X_n - K + 2P)/S \rfloor + 1$$

Where S denotes the stride. Increasing the stride results in a reduction in the output data size; however, it also reduces the feature extraction because the kernels process less data.

After a convolution layer, it is common to implement a pooling layer. The main purpose of pooling is to reduce the size of the data, which also decreases the network parameters and computational load. Pooling merges contiguous data into one value, and the resulting value can be computed by different means, the most common approach is to take the highest value, known as max pooling. Other approaches consider the average, L^2 norm or median. Square windows are typically used for pooling.

2.5 Transfer learning

Formally, transfer learning is defined in [Pan and Yang, 2010, He and Wu, 2017] as follows: given a source domain D_S, a learning task T_S, a target domain D_T and a learning task T_T, transfer learning aims to improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.

A domain D is defined as a pair $D = \{\mathscr{X}, P(X)\}$. Thus, condition $D_S \neq D_T$ implies that either $\mathscr{X}_{\mathscr{S}} \neq \mathscr{X}_{\mathscr{T}}$ or $P_S(X) \neq P_T(x)$, where $P(X)$ is the marginal distribution of X. Similarly, a task is defined as a pair $T = \{\mathscr{Y}, P(Y|X)\}$ and the condition $T_S \neq T_T$ follows the same criteria as domain D.

Following the descriptions in [Pan and Yang, 2010], transfer learning can be divided into three categories depending on the approach.

Inductive Transfer Learning: It aims to improve the learning of a target predictive function $f_T(\cdot)$ in D_T using knowledge in D_S and T_S, where $T_S \neq T_T$. In this case, the target task is different from the source task, despite the source and target domains are the same or not. Labeled data in the target domain are required to induce an objective predictive function.

Transductive Transfer Learning: It aims to improve the learning of a target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$ and $T_S = T_T$. This approach does not require labeled data in the target domain; however, labeled data in the source domain are available.

Unsupervised Transfer Learning: It aims to improve the learning of a target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $T_S \neq T_T$ and Y_S and Y_T are not observable. This case is similar to inductive learning: the target task is different from, but related to the source task. However, it focuses on solving unsupervised learning tasks within the target domain.

In [Pan and Yang, 2010], four transfer learning approaches were defined depending on what is being transferred.

Instance transfer: It assumes that certain parts of the data in the source domain can be reused for learning the target domain by re-weighting.

Feature representation transfer: The knowledge transferred is encoded into a feature representation of the target domain. This representation can improve the target performance.

Parameter transfer: It assumes that source and target tasks share parameters or prior distributions of hyper-parameters. The transferred knowledge is encoded into the shared parameters or priors.

Relation knowledge transfer: In this case, it is assumed that a relationship between the source and target domain data exists. Thus, the knowledge to be transferred is the relationship between the data. In this context, statistical relational learning techniques have been proposed.

2.5.1 Incremental learning

Currently, with a lack of consensus, slight changes in machine learning algorithms receive different names. Incremental, class incremental, life-long, online, never-ending and evolutionary are terms used for specific cases of the same problem. For example, according to the definitions in [Gepperth and Hammer, 2016], life-long learning is a continuous model adaptation method based on constantly arriving data streams. Online learning is applied when training examples are provided only once in the model instead of iterating across training sessions. Incremental learning is a machine learning paradigm in which the learning process takes place whenever new examples emerge and adjust previous learning [Ade and Deshmukh, 2013]. In [Polikar et al., 2001], incremental learning was defined as an algorithm that fulfills the following tasks:

- Ability to learn additional information from new data.
- Have no access to the original data used to train the classifier.
- Preserve previously acquired knowledge.
- Ability to incorporate new classes.

Other studies, such as [Castro et al., 2018, Ade and Deshmukh, 2013], add other criteria to the incremental learning definition, such as end-to-end architecture, limited processing and memory resources. Nevertheless, a formal definition has not been established yet.

However, all these previous terms are specific cases of transfer learning. According to the former definitions, Incremental Learning is a special case of Inductive transfer learning.

A special case of Inductive Transfer learning that aims to increase the classes of a trained model is usually named Incremental Class Learning [Tao et al., 2020, Masana et al., 2020]. There are different categorizations for Incremental Class Learning Approaches, [Tao et al., 2020] recognize three categories:

- **Rehearsal approaches.** Aim to store exemplars of old data to reduce forgetting.
- **Architectural approaches.** Manipulate the network to include new classes.
- **Regularization approaches.** Set regularization to the parameters, losses or inputs of the network.

Furthermore, [Masana et al., 2020] defined the following categories:

- **Regularization based.** Aims to minimize the impact of new data on weights that are important for old data.
- **Exemplar based.** Stores exemplars of old data to prevent forgetting.
- **Task recency bias.** Refers to correcting bias towards new data.

One of the first approaches to address Incremental Class Learning was *Knowledge Distillation*, which was designed to learn a compact student network from a larger teacher network [Hinton et al., 2015]. This approach usually retains information from old classes, i.e., exemplars. Nevertheless, this leads to an imbalance class problem due to the instances from the old classes are not equal to those of the new classes. The basic principle of distillation is to transfer knowledge to the distilled model by feeding a pre-trained model and using the outputs as soft labels, then the student network is trained to learn the behavior of the teacher network, see Fig. 2.4.

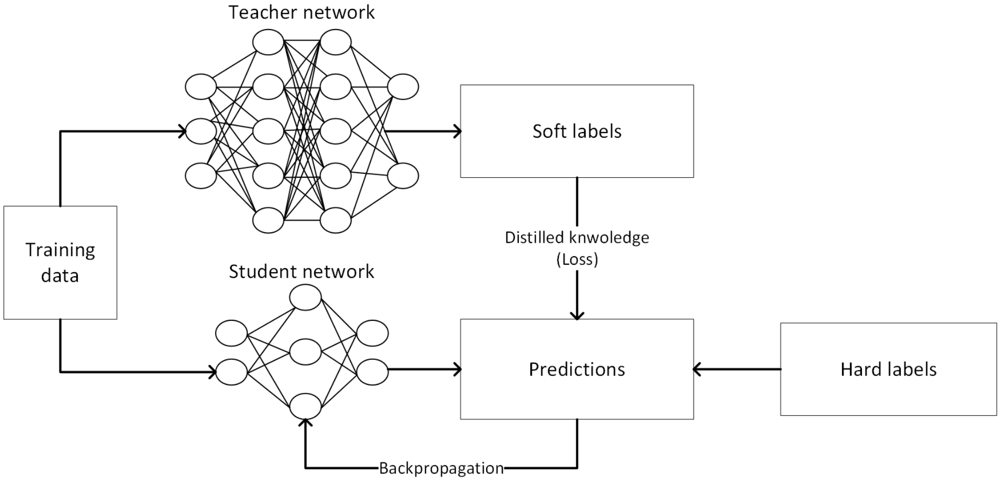


Figure 2.4: Knowledge distillation scheme.

The most relevant issues of these approaches are *Catastrophic Forgetting* which is a drastic decrease in the performance over the old classes, and *Intransigence* which inhibits adaptation of the new class [Chaudhry et al., 2018]. This is also known as the *Stability-Plasticity Dilemma* [Mermillod et al., 2013]. Some of the studies presented in the following Section have developed methods to address these issues while implementing Incremental Learning.

Chapter 3

Related work

In this Chapter, a revision of related work is presented. The most relevant topics for this study are neurophysiological evidence of imagined speech, actual applications of imagined speech in BCI, research on dictionary learning, neural networks and transfer learning. In addition, a discussion of these studies and their relevance to the present work is summarized.

Three main topics were reviewed, and are grouped as shown in Fig. 3.1.

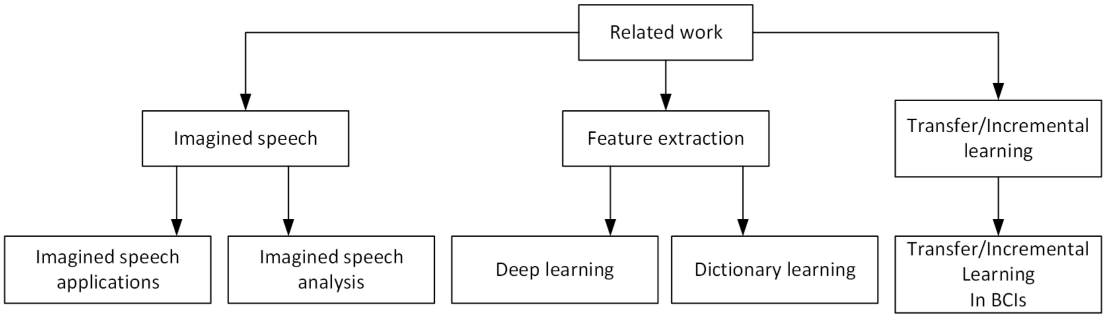


Figure 3.1: Related work taxonomy

3.1 Imagined speech analysis

The Table 3.1 summarizes exploratory research on brain activation areas related to overt and imagined speech. Different acquisition methods have been presented, such as electroencephalography (EEG), positron emission tomography (PET), functional magnetic resonance imaging (fMRI) and electrocorticography (ECoG).

Table 3.1: Imagined speech analysis

Work	**Analysis**	**Results**
[Perani et al., 1996]	PET	The activation areas by speaking a first and a second language show differences in temporal poles. Also, the second language did not activate the same areas as the first language.
[McGuire et al., 1996]	PET	Auditory imagery and imagined speech show similar patterns of decreases of Cerebral Blow Flow in the posterior part of the right middle temporal gyrus, the posterior cingulate cortex and the precuneus. (Areas related to vision).
[Kober et al., 2001]	FMRI	Language processing is localized in the Wernicke and Broca areas. Nevertheless, there is a wide sparse activation in the temporal lobe, the inferior, medium and frontal superior gyrus, the pre and postcentral gyrus and the motor area.
[Shergill et al., 2003]	PET	An inner speech analysis shows activations in the following areas; left superior temporal gyrus, right precentral gyrus, superior temporal gyrus and the adjacent post-central, and bilateral activation of the frontal pole, the parahippocampus gyrus and the bilateral cerebellar cortex.
[Aleman et al., 2005]	FMRI	There is a common activation among speech perception and imagery in the frontal left lobe and temporal lobe. Also, there are convergences in the posterior parietal cortex.
[Liu et al., 2007]	EEG	It is possible to detect neurological differences among semantic categories of imagined words.
[Deng et al., 2013]	EEG	A dipole reconstruction from the heard speech is used to classify imagined speech. The classification of five sentences was performed, reaching an accuracy of 75 percent.
[Wymbs et al., 2013]	FMRI	Overt and covert stuttering utterances, present similar activation areas.
[Xia et al., 2016]	EEG	Semantic plays a role in word recognition on imagined speech. Also, Chinese nouns and verbs are processed differently in the brain. The overall accuracy was 91 percent for binary classification.
[Martin et al., 2016]	ECoG	Imagined speech has few discriminative features. It is attributed to the limited number of sensors, the condition of subjects and the low task comprehension. The accuracy achieved was 57.7 percent for six words.

The use of spatial information from brain signals is relevant to the analysis of imagined speech. However, [Aleman et al., 2005, Deng et al., 2013, Wymbs et al., 2013, McGuire et al., 1996] suggested that focused brain areas are related to imagined speech. However, [Kober et al., 2001, Perani et al., 1996, Shergill et al., 2003] have shown that widespread brain regions are involved in imagined speech. In [Liu et al., 2007], semantic groups were distinguished on the basis of differences in neurological activity. Considering this evidence, the methods proposed in this study consider all available channels.

3.2 Imagined speech applications

Early BCI approaches assumed that users are able to generate specific cerebral signals or take advantage of natural brain responses to external stimuli. An imagined speech based BCI exploits the signals generated by the cognitive process of speech. The main advantage of this approach is that the user does not require special training to generate cerebral signals. The main drawback of imagined speech recognition is the high abstraction level of the cognitive tasks [Brigham and Kumar, 2010].

Different works related to imagined speech BCIs are discussed below. Such studies differ not only in the proposed method but also in the evaluation process. The experimental designs used different subjects, acquisition protocols and imagined speech vocabularies.

The classification of imagined words was formerly reported in [Suppes et al., 1997a], which analyzed EEG and MEG signals to classify seven words (first, second, third, yes, no, right, left), with 100 trials per word for seven subjects. Feature extraction is based on a Fast Fourier Transform (FFT) and a bandpass filter to apply an Inverse Fast Fourier Transform (IFFT). The signals were compared using least squares with a prototype created from their mean, obtaining an accuracy of $52.57 \pm 20\%$ for five subjects.

A simpler scheme was proposed in [Salama et al., 2014b], in which two Arabic words (Yes, No) for seven subjects and 14 trials were classified. The low alpha, high alpha, low beta and high beta rhythms of one EEG channel were analyzed using two methods, the first one obtains statistical data of the signal (minimum, maximum and mean), and the second applies a DWT with six decomposition levels. The classification was performed using Support Vector Machines, Linear Discriminant Analysis, Self Organized Maps, Multilayer Perceptron and assemblies of them, the total mean accuracy was 56%.

Better classification results were obtained in [Kim et al., 2013], where eight Korean monosyllabic words were classified into two semantic classes related to human faces and numbers. Thirty channels and two subjects performing 80 trials were used. The improvement was a spatio-temporal pattern search that obtained a mean accuracy of 92.46% using a Support Vector Machine.

A different approach was presented in [Zhao and Rudzicz, 2015], in which a set of words was labeled according to phonemic and phonological features. The experimental set included EEG, facial, and audio recordings. Many binary classification schemes have been explored, such as vowel-consonant, presence of nasal, presence of bilabial, presence of high-front vowels and presence of high back vowels. The best recognition rate using only EEG recordings was 63.5% and was obtained in the presence of a nasal with an SVM-quad classifier. The data includes sixty-four channels from 12 subjects and one hundred thirty-two trials.

In [Torres-García et al., 2016] different wavelet families and classifiers were explored for multi-class imagined speech classification, the data set contains five Spanish words (”Arriba”, ”Abajo”, ”Izquierda”, ”Derecha”, ”Seleccionar”) with thirty three trials and twenty seven subjects. In addition, automatic channel selection based on fuzzy inference was implemented to reduce the dataset, and an accuracy of $68.18\% \pm 16$ was achieved.

In [Pressel Coretto et al., 2017], a larger vocabulary was presented, it included the Spanish words (”Arriba”, ”Abajo”, ”Izquierda”, ”Derecha”, ”Adelante”, ”Atras”) that were recorded from fifteen subjects performing fifty trials, and applies the method presented on [Torres-García et al., 2013], the obtained accuracy rate was $18.58\% \pm 1.47$. The low accuracy was related to the randomization of the stimuli in the acquisition protocol and the use of basic spectral feature extraction. Nevertheless, the presence of imagined speech discriminative data in the EEG signals was not discarded. An important factor that was not mentioned by the author is the use of fewer channels, i.e. six channels, which were placed in non-relevant areas for speech.

Table 3.2 summarizes the different approaches to imagined speech related work.

Table 3.2: Imagined speech applications

Imagined speech	**Features**
Vowels [DaSalla et al., 2009, Riaz et al., 2014, Matsumoto and Hori, 2014]	Useful for simple tasks. Limited to five classes in Spanish.
Syllables [D'Zmura et al., 2009, Brigham and Kumar, 2010, Deng et al., 2010]	Increases the possible classes. Semantic context is not easy to associate. The acquisition protocols relate syllables to physical activities.
Words [Suppes et al., 1997b, Wang et al., 2013c, Salama et al., 2014a, Zhao and Rudzicz, 2015, Torres-García et al., 2016, Nguyen et al., 2017, Pressel Coretto et al., 2017, García-Salinas et al., 2017]	Increases the classes to an underlying vocabulary. Relates directly words to semantic representations. Deeper analysis of the signal is required.
Semantic groups [Xia et al., 2016, Nguyen et al., 2017]	Includes the analysis of semantics in the imagined speech. Allows ambiguity among words and groups.
Phonemic/phonologic groups [Chi et al., 2011, Kim et al., 2013]	Phonemic/phonologic units are detected in the brain activity. There is no direct relation among groups and semantics.

Using single words, vocabulary can be increased independently of semantic, phonemic or phonological classification. This is useful for BCI applications in which it is assumed that new words can be learned and discriminated.

Previous research has shown that a robust feature extraction is a useful approach for imagined speech discrimination [Zhao and Rudzicz, 2015, Wang et al., 2013a, Salama et al., 2014b, Torres-García et al., 2016, García-Salinas et al., 2017].

3.3 Dictionary learning and Multivariate decompositions

Dictionary learning has been treated using different approaches such as Auto-encoders, Vector Quantization and Sparse representations. This study emphasizes the use of vector quantization and sparse representations, from which the latter is a generalization of the former which provides a higher expressiveness of the models. As their name suggests, auto-encoders encode the features of an input into a set of Artificial Neural Networks coefficients. Table 3.3 presents a comparison of the EEG analysis proposals using these approaches.

Table 3.3: Dictionary learning for EEG analysis

Work	Sparse	Class info	Multi-dimensional	EEG analysis	Transfer learning
[Aharon et al., 2006]	Yes	No	No	No	No
[Zhou et al., 2012]	Yes	Yes	No	Motor imagery	No
[Zhuolin Jiang et al., 2013]	Yes	Yes	No	No	No
[Barthélemy et al., 2013]	Yes	No	Yes	Motor imagery	No
[Gu et al., 2014]	No	Yes	CSP/PSD	No	No
[Ameri et al., 2015]	Yes	Yes	CSP/PSD	Cognitive tasks	No
[Morioka et al., 2015]	Yes	No	No	Visual selection	Yes
[Ameri et al., 2016]	Yes	Yes	CSP/PSD	Motor imagery	No
[Mo et al., 2017]	Yes	Yes	CSP/PSD	Motor imagery	No

One issue in feature extraction for EEG is the inherent loss of information by matricization or vectorization of multidimensional data, which is the transformation of the data into simpler representation spaces. A way to solve this is tensor decomposition methods that were developed to represent multidimensional data as a sum of components that preserve the relations among multiple dimensions. EEG data are time signals recorded simultaneously by multiple sensors in a bi-dimensional representation. Nevertheless, significant features can be obtained by transforming the signals into other spaces, i.e. frequency space. Table 3.4 lists some uses of tensor decomposition in the signal analysis.

A hierarchical relation was defined in [Kiers, 1991], in which it was established that PARAFAC is a constrained version of Tucker decomposition and that Tucker decomposition is a constrained version of Two-way PCA. Thus, an adequate model generated by PARAFAC can be adequately modeled by Tucker, and successively modeled using Two-way PCA [Bro, 1997]. Moreover, the Tucker decomposition generally has no unique solutions [Kolda and Bader, 2009, Cichocki, 2013, Cichocki et al., 2015]. To obtain unique decompositions, certain constraints must be imposed, such as the core sparsity or independence. These constraints increase the model computation time compared to PARAFAC. Subsequently, a dictionary learning approach using PARAFAC decomposition is proposed in this study.

3.4 Deep learning

Most approaches in deep learning for BCIs are focused on motor imagery, the following review will serve as a base to design a deep learning architecture for imagined speech discrimination.

In [Ren and Wu, 2014] it is assumed that convolutional neural networks are more suitable than standard networks due to their capability to process multi-channel structures. The proposed architecture is a two-layer convolutional deep belief network with a filter size of 6 and 4, and a max pooling of 3 and 2, respectively, for each layer. The accuracy of this method in a 4-classes motor imagery dataset was 87.33 ± 1.88.

Table 3.4: Multivariate decompositions

Work	**Tensor decomposition**	**Exemplary applications**	**Dictionary generation**	**Summary**
[Miwakeichi et al., 2004]	PARAFAC	EEG alpha and theta activity detection	No	Introduces the use of components as instances that are able to discriminate between alpha and theta activity.
[Lee et al., 2007]	Non-negative PARAFAC	Mental task EEG classification	No	Nonnegative PARAFAC adapts better to EEG analysis. The components are processed in search of discriminative patterns.
[Zubair and Wang, 2013]	Tucker decomposition	Image reconstruction	Yes	Sparsity in the core tensor compresses the data and reduces the ambiguity in the dictionary components.
[Peng et al., 2014]	Tucker decomposition	Multi-spectral image denoising	Yes	Extends the search of dictionaries which consider the multiple dimensions at once.
[Zhao and Rudzicz, 2015]	Sliced Oriented Decomposition	EEG imaginary movement	No	Slice tensor decomposition shows discriminative and interpretable results of EEG data.
[Quan et al., 2015]	Tucker decomposition	Dynamic texture recognition	Yes	Dictionaries update by means of tensor decomposition, generates a set of sparse dictionaries.
[Barzegaran et al., 2017]	PARAFAC	Alpha rhythm decomposition	No	Provides evidence of the feasibility of PARAFAC for EEG analysis.

The work presented in [Jia et al., 2014] considers the limitation of labeled EEG data by using Restricted Boltzmann Machines. It aims to classify the binary affective state of subjects watching musical videos by grouping classes into like and dislike. Another consideration of this study is the application of semi-supervised learning, i.e., labeled and unlabeled data. The result is presented as the area under the ROC curve of $0.8 \pm .035$.

The previously affective states dataset was tested in [Suwicha et al., 2014], in which the valence and arousal were classified with an accuracy of $53.42 \pm 9.64/52.03 \pm 9.74$ respectively. To achieve this, a neural network with three hidden auto-encoder layers and two softmax layers was proposed. Moreover, previous feature extraction was performed by obtaining five FFT frequency bands. Principal Component Analysis was applied to these features to apply a Covariate Shift Adaptation of Principal Components.

In [An et al., 2014] an individual Deep Belief Network for each channel was proposed, this network was merged with an Ada-boost algorithm. Several layers were tested using Restricted Boltzmann Machines, which resulted in the best performance of an eight-layer network. The input data were processed using FFT and a classification rate of 80.5% was obtained for binary motor imagery.

Different feature extraction methods are applied before the data are fed into neural networks, mainly frequency domain transforms. In [Yang et al., 2015], an augmented Common Spatial Pattern (ACSP) was implemented, which allowed the extraction of CSP features from the multiple level decomposition of frequency.

Other approaches consider spatial information of the signal, e.g. generating activation maps, as in [Bashivan et al., 2016]. This map preserves the spatial structure of the channels and is divided into three frequency bands obtained through FFT decomposition. The time dimension was considered by dividing the signal into segments of 0.5 seconds. This data processing allowed the implementation of a convolutional-recurrent neural network that discriminated four different mental load tasks with a recognition accuracy of 91.11%.

Convolutional networks can be disposed of in different ways, depending on the input data. In [Tang et al., 2017], two one-dimensional convolution layers were proposed, the first into the spatial dimension, and the second into the time dimension. Finally, a fully connected layer and two neuron layers were set. This method aims to find spatiotemporal features in the raw EEG signal of a binary motor imagery task and obtained an accuracy of 86.41 ± 0.77.

In [Wang et al., 2017] imagined speech discrimination was proposed, this approach considers two imagined vowels (/a/,/u/) and a rest state. From this set, three combinations were created: /a/-/u/, /a/-rest and /u/-rest. Subsequently, random sequences from them were generated. The network architecture has three convolutional layers: a one-dimensional convolutional layer that fuses channels (spatial dimension), a bi-dimensional convolution (spatial-temporal), and a bi-dimensional convolution. The signal was fed as temporal segments, and a recurrent layer was also included. The results indicate a recognition loss of 2.1.

A combination of previous approaches can be seen in [Shen et al., 2017]. This study proposed a single network per EEG channel that contains two one-dimensional convolutional layers, followed by a recurrent LSTM layer, a fully connected layer, and a softmax layer, which are merged into a max-pooling layer that determines the predicted label. This architecture achieves an accuracy of 68.51%.

A different approach in [Kaushik et al., 2019] aims to discriminate among age ranges and gender of subjects. The DWT was applied, and a deep recurrent network was proposed. The first layer is a bi-directional Long-Short Term Memory followed by two Long-Short Term Memory layers and two fully connected layers. This approach achieved an accuracy of 93.69% for six age range classification and 97.52% for gender classification.

In [Tayeb et al., 2019], an EEG signal is treated as an ”image-like representation” by applying an STFT. A pragmatic Convolutional Neural Network was proposed, the data were treated as a three-dimensional representation, and three layers of bi-dimensional convolutions were set. This achieved an accuracy of 84.24% for the two motor imagery tasks.

Following the previous approach, [Zhang et al., 2019] considered a three dimensional approach of data. In this case, two bi-dimensional convolutional layers are set, followed by a fully connected layer, and finally a prediction layer for binary motor imagery. In this case, DWT was applied, and the obtained coefficients were reshaped into a 32 by 32 matrix.

Table 3.5 summarizes the network architectures for different EEG analyses.

Table 3.5: Deep learning architectures in EEG

Work	**Analysis**	**Feature extraction**	**Network**	**Result**
[Ren and Wu, 2014]	Motor imagery 4 classes	FFT/PCA	Convolutional deep belief	87.33 ± 1.88
[Suwicha et al., 2014]	Affective states binary	PSD/PCA	DBN Stacked Auto-encoders	$53.42 \pm 9.64/52.03 \pm 9.74$
[An et al., 2014]	Motor imagery binary	FFT	DBN Restricted Boltzmann Machines	80.5
[Yang et al., 2015]	Motor imagery 4 classes	Augmented CSP	Convolutional	69.27
[Bashivan et al., 2016]	Mental load 4 classes	FFT	Convolutional / Recurrent	91.11
[Lu et al., 2017]	Motor imagery binary	FFT/WPD	DBN Restricted Boltzmann Machines	84
[Tang et al., 2017]	Motor imagery binary	FFT	Convolutional	86.41 ± 0.77
[Tibor et al., 2017]	Motor imagery 4 classes	-	Convolutional	92.4
[Wang et al., 2017]	Imagined speech 3 classes	CSP	Convolutional	97.9
[Zhang et al., 2017]	Motor imagery binary	STFT	Convolutional	92.73
[Kumar et al., 2017]	Motor imagery 3 classes	CSP	DBN Auto-encoders	$\sim$88
[Shen et al., 2017]	Motor imagery binary + Mental task	-	Convolutional / Recurrent	68.51
[Tabar, 2017]	Motor imagery binary	STFT	Convolutional / Auto-encoders	90
[Tayeb et al., 2019]	Motor imagery binary	STFT	Convolutional / Recurrent	92.8 ± 1.69
[Zhang et al., 2019]	Motor imagery binary	WPD	Convolutional	78.2

Most previous studies agree that a frequency domain transform of the data provides better discrimination performance when combined with deep learning approaches. Once the data are transformed into a bi-dimensional or three-dimensional set, convolutional approaches are proposed, and the underlying idea is to reduce the data size while increasing the feature extraction for deeper layers. Other approaches take advantage of signal temporal features by applying segmentation and using recurrent networks. Both, frequency and temporal features are well studied in such studies; nevertheless, the spatial dimension is neglected in the analysis. Few studies have considered this approach for feature analysis.

3.5 Transfer/Incremental learning

Incremental transfer learning has been proposed over different machine learning approaches. In [Gepperth and Hammer, 2016] different incremental learning approaches were defined:

- Support vector machines and generalized linear models
- Connectionist models
- Explicit partitioning approaches
- Ensemble methods
- Prototype based methods

Incremental learning has been widely developed for connectionist models such as neural networks. In [Li and Hoiem, 2018], some approaches were grouped into categories, the three most representative of which are Fine-tuning, Feature extraction and Joint Training. In this study, a new approach called Learning without Forgetting was proposed, which can be considered in the feature extraction category.

In [Polikar et al., 2001], one of the former algorithms for incremental learning neural networks was proposed. It is based on an ensemble of simple neural networks (NNs). However, a classifier can also be used in the ensemble.

Image analysis is an incremental learning primary field. [Bart and Ullman, 2005] presented an approach based on images patches and the classes were represented by a bag of features. These patches were correlated with the images to search for similarities considering the spatial location and likelihood ratio. It is necessary to define novel fragments when a new class arrives.

Hierarchical incremental networks are a viable solution for incremental learning, in [Xiao et al., 2014] a hierarchical model in combination with neural networks was shown. The main idea is to group the classes into super-classes that can be split when new data are fed. A specific classifier is trained for each superclass. The main drawback is defining a similarity measure to merge or split super-classes. In [Roy et al., 2020], super-class networks were grown as trees when new classes were fed. The super-class network evaluates the inputs and determines which sub-class of the network corresponds to. Subsequently, a branch network performs a sharp classification. In [Sarwar et al., 2020], a tree approach was proposed, in this case, the branches correspond to old and new data, and share a base network. The new branches are trained independently and added to the old branches to update the network.

Prototype-based incremental learning was presented in [Rebuffi et al., 2017], the main drawback of which is the use of *exemplar images* as well as class prototypes. Nevertheless, it sets the basis for further studies, such as [Cheng et al., 2019], in which the last layer of a neural network is transformed into a Nearest Class Mean (NCM) classifier, which is a special case of k-Nearest Neighbors. This layer can add a new class by creating the mean of new classes. Finally, classification was performed using a probability softmax function that assigned the input vector to the closest mean.

In [Ye and Zhu, 2019], there is an interest for implementing a function that reduces the intra-class and increases the inter-class distance of the network outputs. Using these improved distances, an SVM-based classifier was applied to the old and new classes. Following the previous idea, the outputs of any neural network can be used as features for other machine learning methods. In [Hasan and Roy-Chowdhury, 2014], an ensemble of SVM classifiers was proposed based on the previous study.

The use of pre-trained neural network models for new tasks resulted in a term called *knowledge distillation*, in which the information of a cumbersome network is transferred to a lighter network. [Zhang et al., 2020] improved the distillation for incremental learning. However, this proposal requires auxiliary data in the training step, also known as exemplar data from the old classes.

In [Hao et al., 2019], the information obtained from old classes is retained by a teacher network and it is distilled to a student network that learns the new incremental classes using only information of these new classes. In addition, the use of a prototype-based classifier was proposed to retain the information of old classes.

In some cases, only are a small number of instances of the new class is available, [Tao et al., 2020] named this approach "Few shot Incremental Learning", and established that knowledge distillation presents some issues to address this approach as the class imbalance and the performance trade-off across new and old classes. The use of pre-trained neural network models for new tasks resulted in a term called *knowledge distillation*, in which the information of a cumbersome network is transferred to a lighter network. [Zhang et al., 2020] improved the distillation for incremental learning. However, this proposal requires auxiliary data in the training step, also known as exemplar data from old classes.

In [Hao et al., 2019], the information obtained from old classes is retained by a teacher network and it is distilled to a student network that learns the new incremental classes using only information of these new classes. In addition, the use of a prototype-based classifier was proposed to retain the information of old classes.

In [Masana et al., 2020], different incremental approaches were tested and compared, and it was concluded that exemplar-based methods cannot compete with exemplar-free methods. Moreover, network architecture results may vary depending on the case of the study. Nevertheless, in most of the results, the incremental models only mitigate forgetting instead of increasing or even maintaining the performance. Good results were obtained with small domain shifts with a large number of samples.

3.6 Transfer/Incremental learning in BCIs

Due to the variability in brain signals across subjects, or sessions for the same subject [Morioka et al., 2015], transfer learning provides an area of opportunity for BCIs. Most transfer approaches are based on Common Spatial Patterns (CSPs), assuming that a set of invariant filters exists across sessions or subjects [Jayaram et al., 2016].

A different approach on CSP [Kang et al., 2009], proposed a combination of the CSP covariance matrices of different subjects for binary motor imagery classification. Emphasis on similar subjects was proposed by measuring the divergence between data distributions.

The work presented in [Wu et al., 2014] merged transfer learning with active learning to classify visually evoked potentials in EEGs. Active learning allows for the selection of the most relevant data. However, estimating the most informative data required additional processing.

The resting state of the brain is analyzed by [Morioka et al., 2015] assuming that it reflects the subject-specific nature of brain activity. Thus, the resting state of a subject was used to calibrate a previously learned model.

A deeper analysis of the signal was performed in [Wronkiewicz et al., 2015] for subject transfer learning, in this work the T1-weighted Magnetic Resonance Imaging (MRI) is considered in addition to EEG recordings. The underlying idea is that training a model using anatomical data from MRI may compensate for the variability in electrode positioning and head morphology, which may improve transfer learning.

In [Panagopoulos, 2017], in addition to transfer learning, the use of a low cost and user-friendly EEG device was proposed. Moreover, the feature extraction methods were based on frequency band extraction and raw signal analysis was performed.

There is an area of interest in the search of common brain activity patterns across subjects. Nevertheless, some approaches do not assume that a similar pattern exists between subjects. This is the case in [Dalhoumi et al., 2014], where a classifier was built for each subject and the transfer was performed by adjusting these classifiers for a new user.

In [Tu and Sun, 2012], a spatial filter bank generation per subject was proposed, such filters are assembled following selection criteria for motor imagery classification. The subject transfer was performed by selecting filter training sets of subjects whose features were similar to those of the target user. A drawback of using this method is the definition of relevant features for different subjects.

Table 3.6 summarizes relevant studies that aimed to transfer learning applied to a BCI system.

Table 3.6: Transfer learning in BCIs

Work	Transferring	Features	Task
[Kang et al., 2009]	Subject	CSP covariance	Binary motor imagery
[Tu and Sun, 2012]	Subject	Spatial filter banks	Binary motor imagery
[Wu et al., 2014]	Subject	Raw magnitudes	Binary visually evoked potentials
[Dalhoumi et al., 2014]	Subject	CSP	Multi-task motor imagery
[Lotte, 2015]	Subject	CSP / LDA	Motor imagery / Workload / Mental imagery
[Wronkiewicz et al., 2015]	Subject	Spatial normalization	Auditory attentional switching
[Morioka et al., 2015]	Subject	CSP dictionaries	Visual spatial attention
[Jayaram et al., 2016]	Subject / Sessions	Spatial-Spectral features	Binary motor imagery
[Waytowich et al., 2016]	Subject	Spectral ensemble / Information geometry	Binary visually evoked potentials
[Panagopoulos, 2017]	Subject	Logistic regression / Bayesian inference	Multiple cognitive tasks
[He and Wu, 2017]	Subject	CSP	Multi-task motor imagery
[Wei et al., 2018]	Subject	Power spectrum clustering	Drowsiness detection

Most transfer approaches in BCIs are oriented toward time suppression for the calibration of new subjects, i.e. transferring a model without requiring data from the new subject. Two categories were presented by [Lotte, 2015], Pooled design, which optimizes a single model on the combined data of multiple users. Ensemble design, in which a model is optimized for each user and combined afterward.

In this work, by using transfer learning, the possibility to extend a generated imagined speech model to new imagined words is proposed, this can be seen as an inter-subject transfer learning task. BCI transfer is commonly focused on intra-subject variability because it is not possible to add many motor imagery commands. Imagined speech may allow a model increase for one subject by including a wider context vocabulary of BCI commands.

3.7 Discussion

Research on overt and imagined speech has found that despite the language region of the brain (Wernicke's area) shows high activation, many regions of the brain are activated during speech tasks. Based on this premise, the proposed analysis of EEG signals considers the use of all the available channels in existing databases to obtain as much information as possible.

Some imagined speech approaches are focused on the identification of vowels and syllables. Although the signal analysis of these approaches has high discrimination performance, its main drawback is the limitation of the available vocabulary. The intuitive extension of vowels and syllables is the use of words, this can take advantage of the vocabulary context. However, this approach required a higher level of feature analysis.

One advantage of dictionary learning approaches is their ability to generate a dictionary that contains a set of the most representative features of the signals. Note that the dictionary depends on a previous feature extraction. Most previous studies have transformed multi-dimensional data and lost the relations among such dimensions. In this study, two approaches are explored: multivariate decomposition and deep learning.

In this work, the advantages of dictionary learning and multivariate decomposition will be combined in order to solve two aims, to achieve good discrimination of imagined speech and to improve the extension of a learned vocabulary. The last approach is transfer learning, which is an open problem in the BCI analysis. BCIs are particularly interested in transferring information between subjects, which involves the adaptation of a model to new subjects. Nevertheless, in this study, a different transfer is proposed: the inclusion of a new word in a previously generated model.

Chapter 4

Dictionary learning and Deep learning comparison for imagined speech classification

The objective of this chapter is to compare two different methods, dictionary learning and deep learning, for imagined speech classification. By comparing their advantages, limitations and performances, one of these methods was adapted to include an incremental learning scheme for imagined speech.

Dictionary learning was chosen for this experiment due to its ability to obtain a set of atoms that represent signals which can lead to the interpretation of brain signals related to each imagined word. On the other hand, the deep learning approach was chosen for this experiment because to its capability of automatic feature extraction, which enhances classification with less pre-processing of the signal. Another difference is related to data processing: neural networks apply feature extraction across the architecture, whereas dictionary learning requires previous feature extraction of the signal.

Moreover, the implementation and computational details are considered in the discussion for wider comparison. The method that presents better advantages in classifying imagined speech can be further improved by adding new words using an incremental learning approach.

4.1 Datasets

Each dataset used in subsequent experiments had different features and acquisition protocols, as described in this section. Thus, the proposed method was tested under different conditions. During the work of this thesis, a new dataset for imagined speech was recorded. A novelty in this dataset is that the brain signal was recorded with EEG and fNIRS devices. In addition, overt speech recordings were also included.

1. The first dataset is obtained from [Torres-García et al., 2016]. The EEG of twenty-seven native Spanish speaking subjects was recorded from 14 channels at a 128 Hz sample rate. The data consists of five imagined speech Spanish words “Arriba”, “Abajo”, “Izquierda”, “Derecha”, “Seleccionar” (translated in English as “Up”, “Down”, “Left”, “Right”, “Select”), repeated thirty three times each one, with a rest period between repetitions. The recordings were performed in a controlled environment without any sound or visual noise. However, in the acquisition protocol, words were presented sequentially.

2. The second dataset is presented in [Nguyen et al., 2017] two approaches are presented, three short words (“In”, “Out, “Up”) and two long words (“Cooperate”, “Independent”). For each task, six subjects were recorded with one-hundred trials per word. These signals were recorded using a BrainProducts ActiCHamp with 64 channels at 1000 Hz. The data were also pre-processed using a band-pass filter between 8 and 70 Hz, a notch filter at 60 Hz and an electro-oculogram artifact removal algorithm.

3. Finally, a dataset was acquired for this work using a g.HiAmp EEG recorder with 31 channels at a 256 Hz sample rate. Six healthy subjects were recorded imagining three Spanish words, “Arriba”, “Abajo”, “Seleccionar”, (translated in English as “Up”, “Down”,“ Select”). The protocol included a small training phase that used five words to ensure that participants were adequate for the task. The words appeared randomly on the screen for 1.5 seconds, in which the subject had to imagine such a word once. Between each word, a rest period of 0.5 seconds is available, in which the screen was turned black.

4.2 Dictionary learning analysis

The proposed method is illustrated in Fig. 4.1, this process is applied to each epoch, i.e. repetition, of the imagined words per subject. The EEG signals are converted to the frequency domain, and later, PARAFAC analysis is applied to obtain a set of components that will be used as inputs for dictionary generation and further classification.

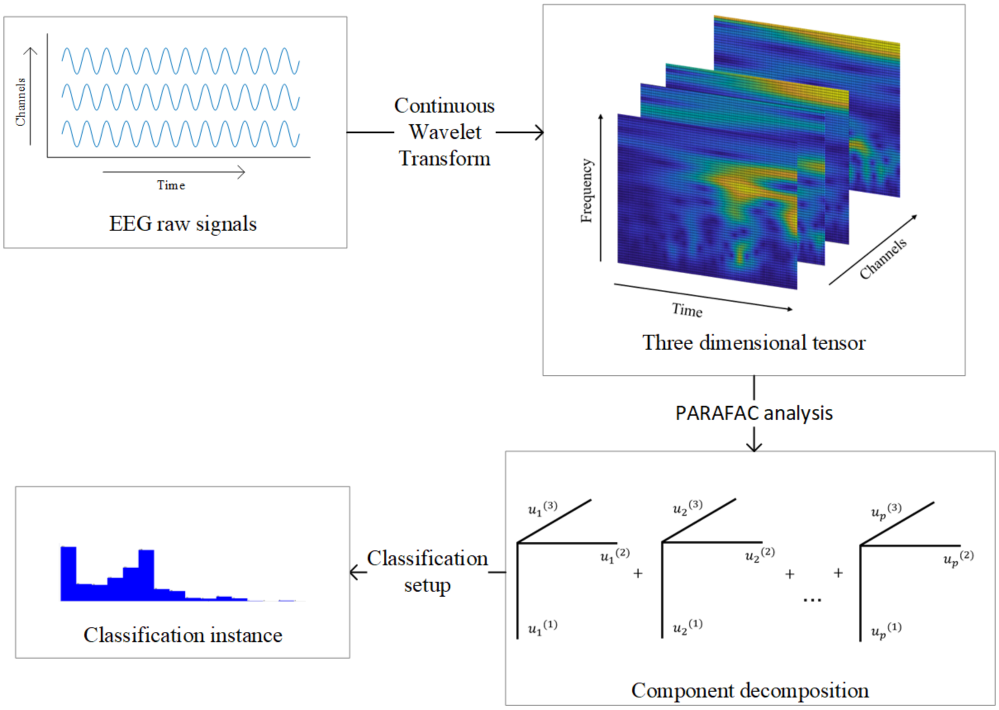


Figure 4.1: Feature extraction for one epoch.

A Continuous Wavelet Transform (CWT) was applied to each epoch of a subject to generate a three-dimensional tensor. PARAFAC analysis was then applied and the obtained components were ordered and labeled to the corresponding imagined word. The components are concatenated by mode, resulting in an element-wise correspondence. This process was repeated for each epoch. Finally, a clustering method is applied to the components to obtain representative prototypes of the data. These prototypes were used to represent the original set of components as a histogram.

A classification was then applied using the extracted components in U. For each mode of the tensor, $n \times p$ vectors $u_k^{(n)}$ were extracted and concatenated into a matrix $A_{s,C_i,e}$ (one matrix for each subject s, class C_i, and epoch e). The components are generated while preserving the information on the relationships between the different modes.

$$A_{s,C_i,e} = \begin{bmatrix} u_1^{(1)} & u_2^{(1)} & \dots & u_p^{(1)} \\ u_1^{(2)} & u_2^{(2)} & \dots & u_p^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ u_1^{(n)} & u_2^{(n)} & \cdots & u_p^{(n)} \end{bmatrix}$$

Each factor $u_k^{(n)}$ is a vector with the form

$$u_k^{(n)} = \begin{bmatrix} u_k^{(n)}(I_1) \\ u_k^{(n)}(I_2) \\ \vdots \\ u_k^{(n)}(I_m) \end{bmatrix}$$

Where I_m denotes the elements of the component vector in mode n.

Subsequently, for each subject, the epoch matrices $A_{s,C_i,e}$ corresponding to the same class C_i were concatenated to apply the k-means clustering algorithm. This procedure was repeated for all classes. Once each class has an associated cluster, they are concatenated to achieve a global representation D_s of the signals for the s subject.

The next step is to analyze the elements of every matrix $A_{s,C_i,e}$ and use the Euclidean distance to find and replace each row with the closest prototype in the global representation D_s. This process transforms the matrix A into a sequence of elements D_s. This sequence of elements is later transformed into a histogram associated with class C_i and becomes a classification instance. Histogram generation converts each epoch e of the matrix $A_{s,C_i,e}$, into a single histogram. Finally, a linear classifier was applied to the set of histograms for each subject. The classifiers were evaluated using a 10-fold cross-validation strategy. This process is summarized in Fig. 4.2.

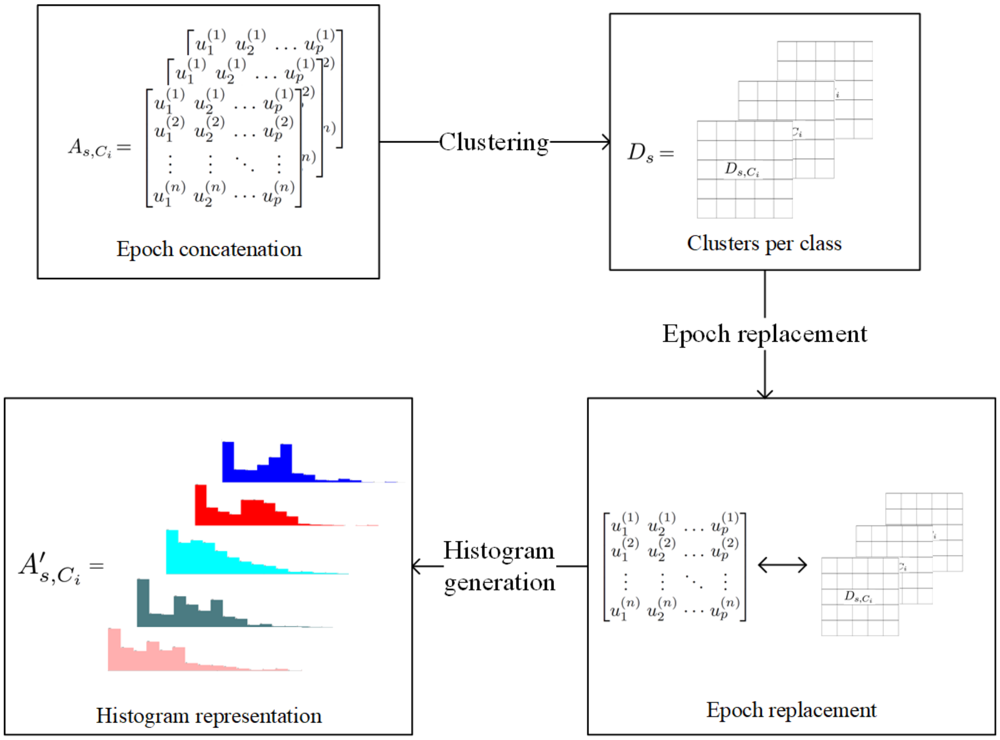


Figure 4.2: Classification setup for one subject.

4.3 Deep learning analysis

Different neural network architectures have been developed over the years. Despite the existence of networks created for specific tasks, a standard for their use has not yet been established. Thus, the objective of the following experiments was to test different architectures that presented good results for brain signal analysis and BCIs. Another consideration was to keep the network architecture as simple as possible to reduce computational costs.

4.3.1 Bi-dimensional approach

In Fig. 4.3 a bi-dimensional approach of the signal is presented. Frequency feature extraction was applied to EEG signals, i.e., Continuous Wavelet Transform. The extracted features from each channel were concatenated into a single matrix (bi-dimensional tensor). Then, convolution in the time dimension was applied for each frequency band to highlight the features and reduce the data length. Subsequently, a fully connected layer is applied, and finally, a soft-max layer predicts the classes. This simple neural network approach provides a baseline for comparing the following proposals.

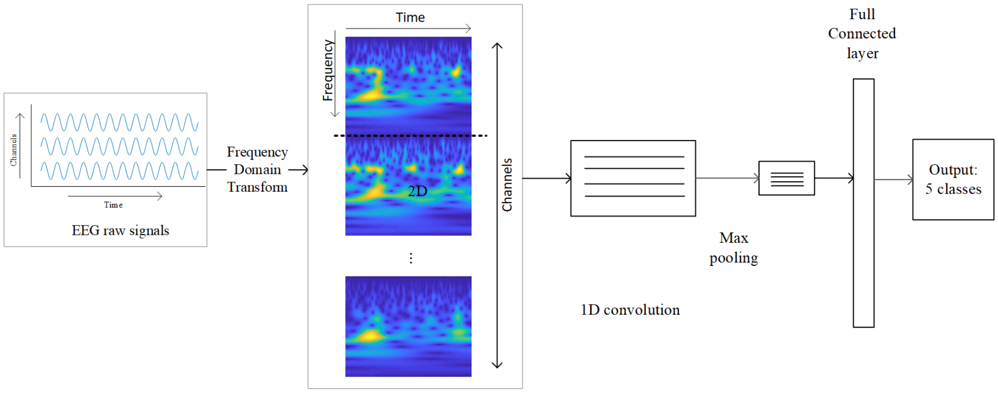


Figure 4.3: Deep learning flowchart

4.3.2 Convolutional network

A frequency domain transform, i.e., Continuous Wavelet Transform, is applied to each channel and a three-dimensional tensor is created by concatenating them into a third dimension. This tensor is processed by a bi-dimensional convolution as shown in Fig. 4.4. It is expected that a bi-dimensional convolution that considers the events of every channel in each time sample and frequency band will be able to extract more relevant features to discriminate imagined speech. The obtained features were later reduced by a max pool layer and then fed into a fully connected layer. Finally, a soft-max layer predicts the classes.

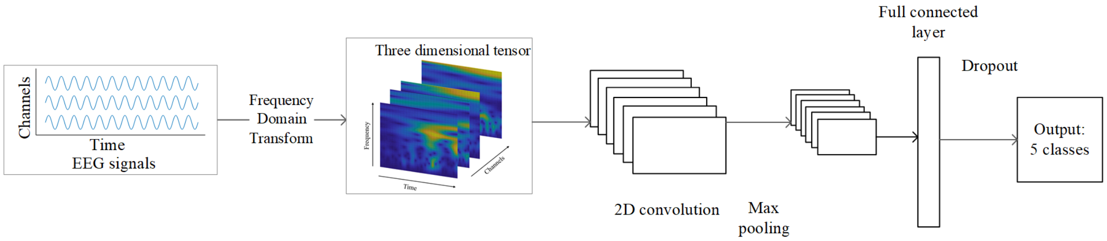


Figure 4.4: Convolutional deep learning flowchart

4.3.3 Long-Short Term Memory network

Considering that EEG signals are naturally represented in the time dimension, the following approach applied a time segmentation to the signal. The first consideration was to create independent processes for each EEG channel. Then, for each channel, the frequency bands segmented over time were fed into an LSTM layer. Subsequently, the results were fed into a fully connected layer to predict the classes with a soft-max layer, as shown in Fig. 4.5.

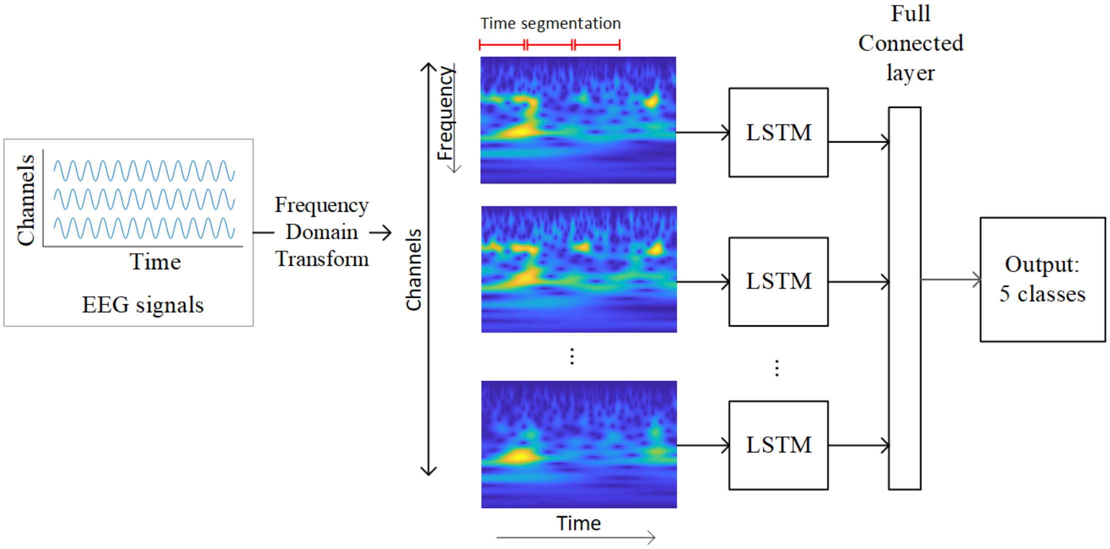


Figure 4.5: Convolutional deep learning flowchart

4.3.4 Convolutional Long-Short Term Memory network

The following approach considers independent processes for each channel. This is an improvement over the previous approach because it includes the convolution of frequency bands in the time dimension. Time segmentation was performed over the signal to feed it into an LSTM layer. This produces a combination of convolutional and LSTM layers. Later on, a fully connected layer is fed, and predictions are made using a soft-max layer, as shown in Fig. 4.6.

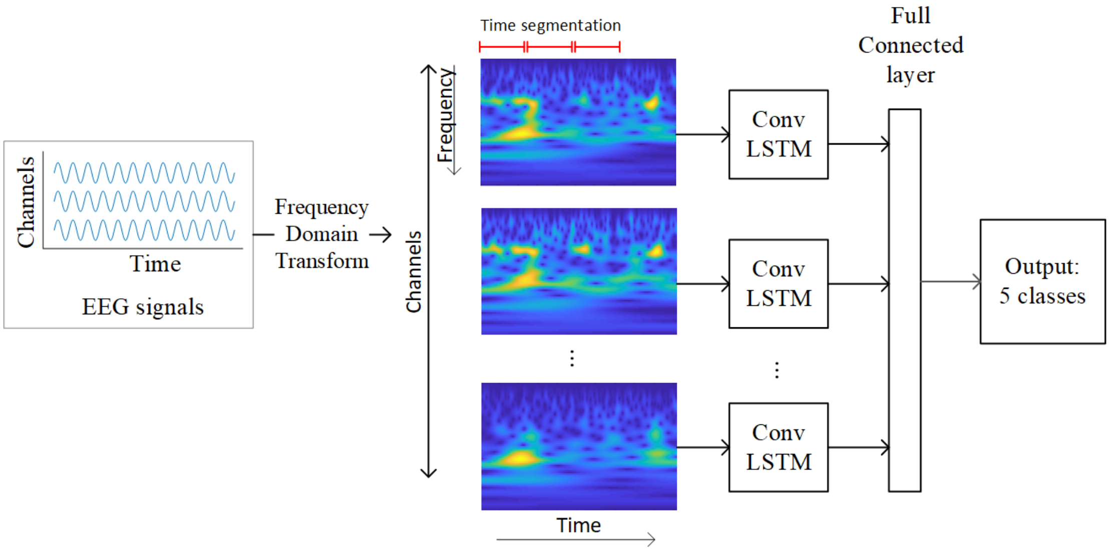


Figure 4.6: Convolutional deep learning flowchart

4.4 Results

The results section is divided into two subsections. First, the results for different proposed architectures are presented. The latter compares dictionary learning and deep learning methods.

4.4.1 Network architectures and parameters

The first experiments for neural networks were aimed at testing different inputs of the signal with the previous feature extraction. In Fig. 4.7, the comparison of different feature extractions of the EEG signal over a convolutional neural network is presented. The different architectures were compared in terms of their accuracy for subjects of the [Torres-García et al., 2016] dataset. The results presented below are the average values for all subjects.

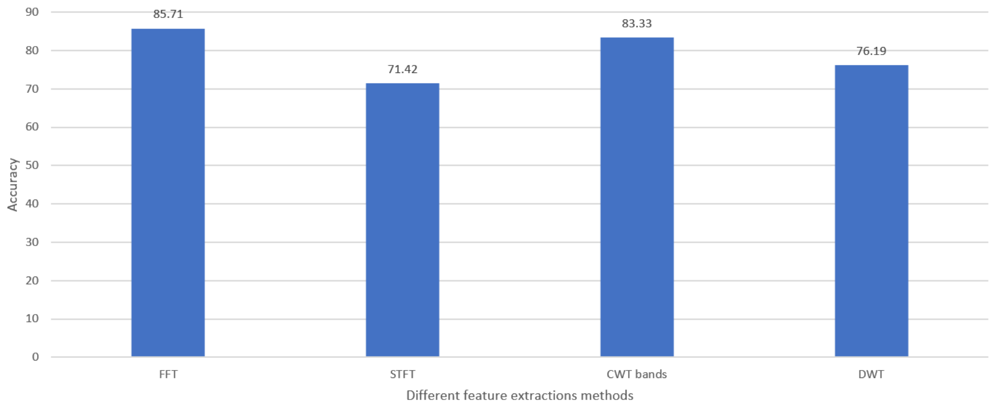


Figure 4.7: Feature extraction comparison

The results for the different network architectures are shown in Fig. 4.8. This comparison was performed to determine the best architecture for fitting data. These results represent the average accuracies of subjects from the [Torres-García et al., 2016] dataset.

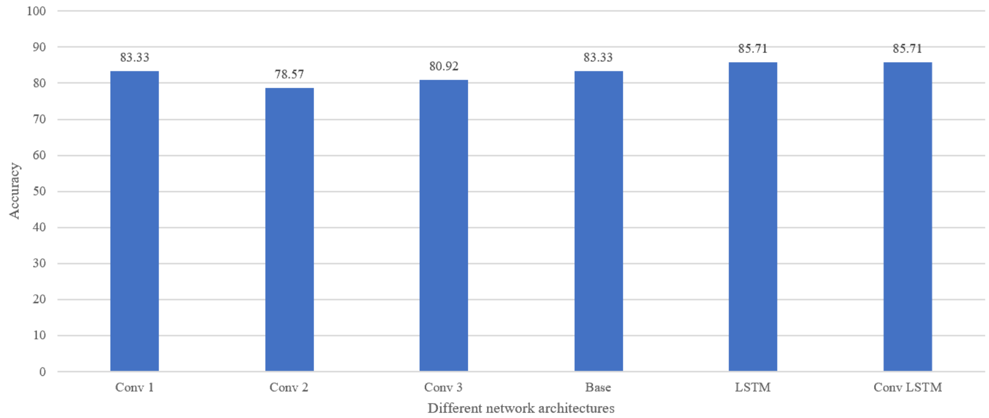


Figure 4.8: Neural network architectures comparison

4.4.2 Dictionary learning and Deep learning

Although different network architectures have shown similar results, the following experiments were performed using a bi-dimensional convolution network, which is adequate for processing multidimensional EEG data. Finally, the baseline results were compared with those of the proposed approaches. The following results were obtained by repeating the procedure five times for each subject, and the average results are presented.

In Fig. 4.9, results from [Torres-García et al., 2016] dataset are presented, in color blue the deep learning architecture, in orange the dictionary learning and the baseline results in gray. It is an average accuracy of 53.21 ± 9.9 using deep learning and 61.42 ± 15.76 using dictionaries, compared with the baseline of 70.11 ± 15.78 from the results in [Torres-García et al., 2016]. The original dataset was obtained from 27 subjects, of which 3 were removed due to technical problems during acquisition. An ANOVA statistical analysis of the deep learning and dictionary learning results showed a difference of $F[1,46] = 4.65, p = 0.036$.

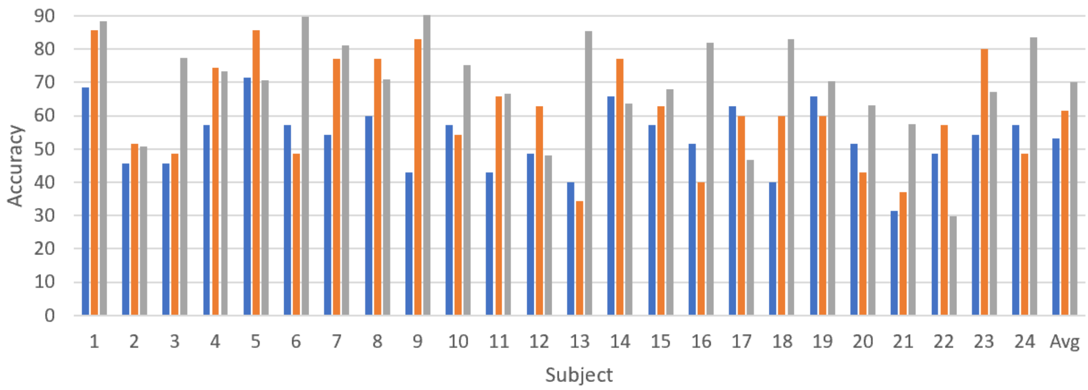


Figure 4.9: [Torres-García et al., 2016] database, in color blue the deep learning architecture, in orange the dictionary learning approach and in gray the [Torres-García et al., 2016] results.

Fig. 4.10 shows the results obtained in [Nguyen et al., 2017] database. The results of the deep learning architecture are presented in blue, those of dictionary learning in orange and those of [Nguyen et al., 2017] in gray. This corresponds to the three short word analyses of the database. The average accuracies obtained were 46.66 ± 1.82 using deep learning, 40.27 ± 8 and 50 ± 3.49 using dictionaries. An ANOVA statistical analysis of the deep learning and dictionary learning results showed that there was no difference with $F[1,10] = 2.57, p = 0.139$.

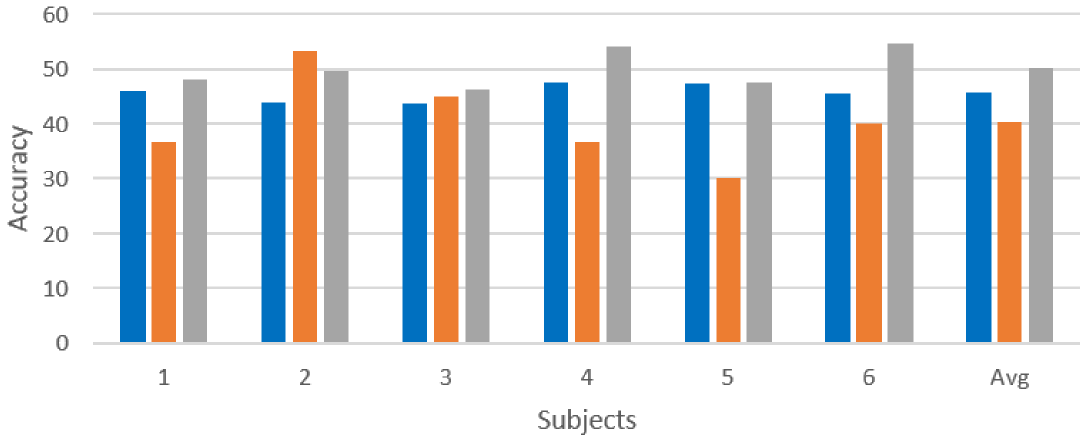


Figure 4.10: [Nguyen et al., 2017] database, in color blue the deep learning architecture, in orange the dictionary learning approach and in gray [Nguyen et al., 2017] results.

Fig. 4.11 shows the results obtained in [Nguyen et al., 2017] database corresponding to the two long word analysis of the database. The results of deep learning architecture are shown in blue, dictionary learning in orange and baseline in gray. In this case, a Fast Fourier Transform was applied to the data. This yielded average accuracies of 69.81 ± 4.12 for the deep learning method and 56.87 ± 9 for the dictionary learning, compared with a baseline of 66.18 ± 4.8 from [Nguyen et al., 2017]. An ANOVA statistical analysis of the deep learning and dictionary learning results showed a difference of $F[1,10] = 10.11, p = 0.0098$.

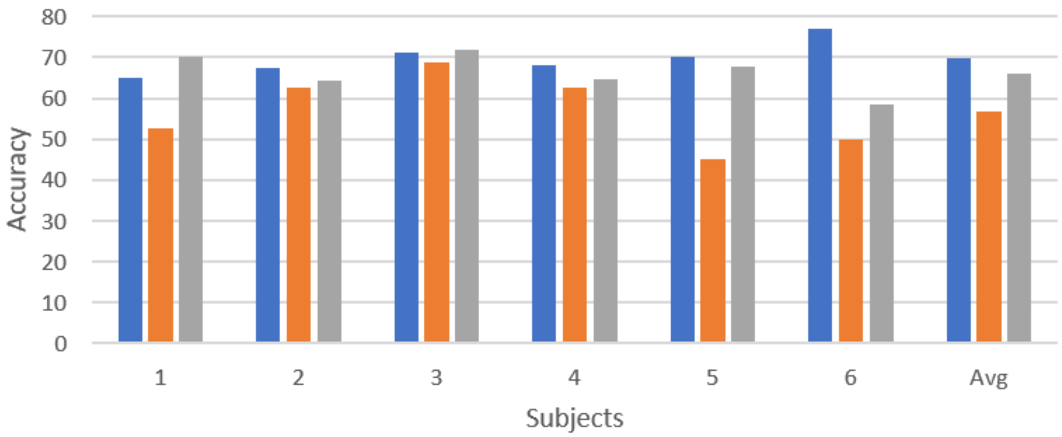


Figure 4.11: [Nguyen et al., 2017] database, in color blue the deep learning architecture, in orange the dictionary learning approach and in gray [Nguyen et al., 2017] results.

The [Nguyen et al., 2017] dataset has a hundred recordings for each word and sixty-four channels, the [Torres-García et al., 2016] dataset has thirty-three repetitions for each word and fourteen channels. This is a huge difference in the available data that could improve the results obtained by the neural networks. In addition the inherent feature extraction that the networks provide aid to improve the classification.

4.5 Discussion

The dictionary learning approach was tested with a Continuous Wavelet Transform that was applied to the data to later apply a Parallel Factor Analysis (PARAFAC) for component extraction. Although the dictionaries based on PARAFAC obtained a good representation performance, the predictions were not promising for the [Torres-García et al., 2016] dataset.

The frequency domain transforms applied to the data affect the recognition performance. Continuous Wavelet Transform (CWT), fast Fourier transform (FFT), short-term Fourier transform (STFT), and discrete wavelet transform (DWT) were applied to the test data. In most cases, the best results were obtained by applying CWT and extracting Band Powers related to brain activity (Low Alpha, High Alpha, Low Beta, High Beta, Gamma, Delta).

The network architectures provided similar results to the different frequency domain transforms, confirming the adaptability of the neural networks to the data inputs. Nevertheless, it must be considered that some transforms are computationally faster and allow faster network training. For one of the databases, the deep networks obtained higher results than the baseline and dictionary methods.

Some disadvantages associated with the use of neural networks include the loss of interpretability and the increase in the parameters of the model. The decision to continue exploring neural networks in further experiments was made because of their flexibility for incremental learning, which is an objective of this study.

Chapter 5

Proposed incremental learning networks

Training a BCI is a costly task that requires time and expert knowledge (e.g., device placement, calibration and data recording), and the user requires considerable effort. Hence, most studies have focused on transferring BCI models from one subject to another. This study also focused on decreasing end-user stress, with the goal of adapting the existing user model by adding new classes, i.e., new commands for the BCI. According to our previous results, deep neural network-based approaches showed good performance for imagined speech classification[1]. Moreover, deep neural networks allow for automatic feature extraction, which facilitates the construction of the final model. This chapter presents two proposed incremental deep learning models for imagined speech.

The original proposal of this study is a neural network architecture for feature extraction which creates a set of centroids to represent classes. Centroids were employed

[1] The decision to continue using the deep neural network approach was also supported by the results of the first experiment using a dictionary-based incremental learning approach. The results of this experiment can be found in Appendix A.

for classification purposes; therefore, the distance between the input data and centroids was the primary parameter used to train the network. Moreover, the proposed architecture grows to allow the inclusion of new classes. When the data of new classes are added, new modules of the network are implemented and trained by considering the information of the original data.

The use of neural networks as feature extractors and centroids has been proposed previously. The novelty of this study is the implementation of multiple centroids, the increase in the network architecture to include new classes and the training of the new architecture considering previous learning.

In this chapter, two incremental learning approaches are proposed. The objective is to add a new class to an existing model without retraining or storing the original data. These approaches were based on the results of the experiments conducted in the previous chapter.

The first approach, named *incremental single network*, aims to train a neural network using only the original data, the architecture is preserved with no changes when a new class is added for incremental learning. Section 5.1 presents details of the method, variations, and results. The second approach, which is presented in detail in Section 5.2, is a neural network improvement that allows the inclusion of a new word by creating independent *twin networks* for new classes.

To compare both approaches, the results from the previous chapter were considered, the experimental procedures were preserved and signal processing was performed using Power Spectrum Density (PSD) to obtain the frequency information from the signal, which was adopted because of its wide use in previous studies [Suwicha et al., 2014, Gu et al., 2014, Ameri et al., 2015, Ameri et al., 2016, Mo et al., 2017, Wei et al., 2018]. Moreover, the PSD method is based on the FFT, which obtained the best results in the previous chapter.

The previous results were similar for different network architectures; thus, the simplest convolutional network was chosen to reduce implementation costs and computation time, as shown in Fig. 5.1.

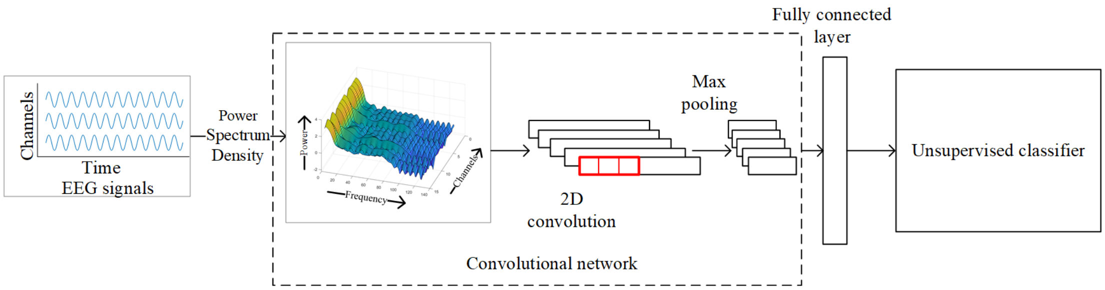


Figure 5.1: Convolutional network design

This convolutional network is configured as shown in Table 5.1, and the input size for the network corresponded to the number of channels in each dataset. The proposed incremental approaches consider a neural network as a feature extractor; thus, the feature vectors for incremental learning are the outputs of a fully connected layer.

Table 5.1: Network parameters

Dataset	[Torres-García et al., 2016]	[Nguyen et al., 2017]	New dataset
Input	(14, 1, 129)	(62, 1, 129)	(31, 1, 129)
Convolution	Kernel size: (1,5), Stride: 1, Filters: 100		
MaxPool	Kernel size: (1,2)		
Fully Connected Layer	1000 neurons		

Finally, for every experiment, the following details were considered:

- For each dataset the data of six subjects were used to fit the same number of subjects.
- Each experiment was repeated five times per subject due to the stochastic behavior of the methods.
- The results are given as the average of the six subjects (five runs for each subject).

- Data were split in 80% for training and 20% for testing purposes, for original and new classes
- The complete training set was always employed for old classes.
- The new class was added to the model using a different number of training instances that started with one and increased by two until the complete training instances were used.
- The incremental approach was tested with the test partition (20% of datasets).

5.1 Incremental single network

The incremental single network approach uses a neural network architecture as a feature extractor and preserves the structure with no changes as new classes are added. This proposal does not increase the computational cost of adding new words because the network is not trained for the new classes. Furthermore, classification is performed by processing the network outputs as feature vectors. Two classification schemes were tested and compared in terms of overall classification accuracy and the added class performance.

5.1.1 Centroid-based incremental network

This method similar to the algorithm proposed in [Cheng et al., 2019], including the convolutional implementation shown in Fig. 5.2.

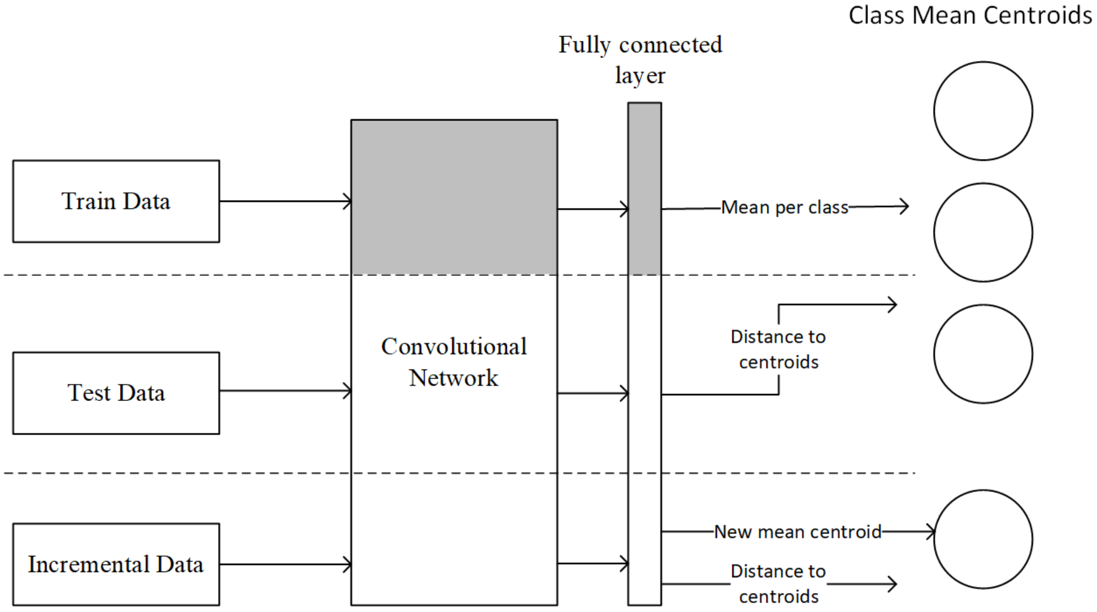


Figure 5.2: Centroid-based incremental network, back-propagation adjust in color grey.

The training data were fed through the network, and in the last layer, a mean centroid for each class is generated using the network outputs. Then, for each instance of the training data, the euclidean distance to each centroid was measured to compute the prediction error. From the obtained error, a loss value was computed and back-propagated to fit the network.

Later on, for each test instance, the distance to every class mean is computed. The instance is then labeled as the class of the closest mean.

For the incremental step, instances of the new class are fed, the network does not retrain the parameters and generates a new centroid for the new class. It is expected that the new class centroid will be sufficiently different from the others, as confirmed by performing the test using data from the original and new classes.

Due to the classification based on a distance measure to one centroid being quite simple, most of the performance relies on the network training. Thus, the next proposed method will consider increasing the pattern recognition of network outputs.

5.1.2 kNN-based incremental network

For this approach, the training data were fed through a convolutional network and the outputs were employed as features for classification. In this case, the network outputs are used as feature vectors for a kNN classifier. This classification does not require training, the instances are transformed into objects over a space, and are related to each other only by their euclidean distances, as shown in Fig. 5.3.

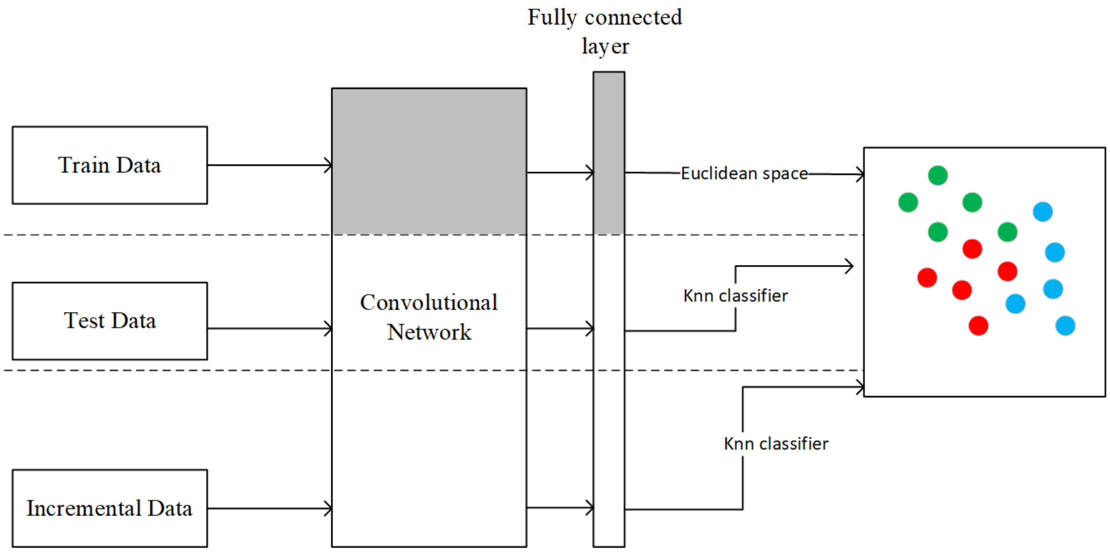


Figure 5.3: kNN-based incremental network, back-propagation adjust in color grey.

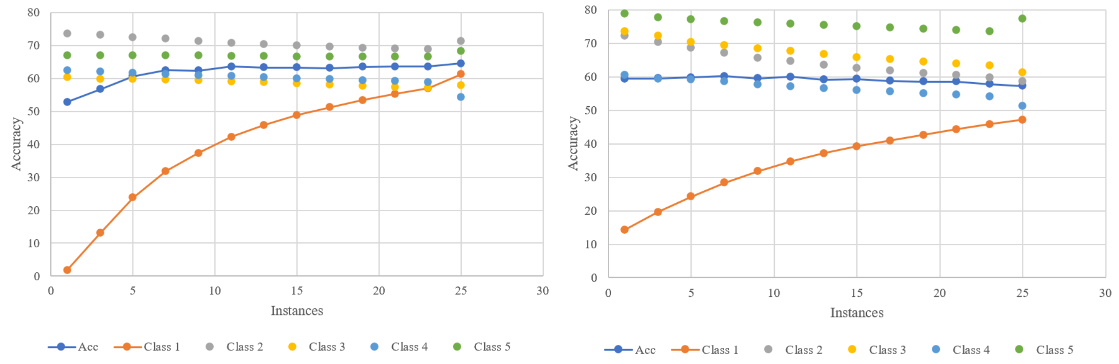
Subsequently, for testing purposes, the instances were fed into the network and the kNN method was applied to measure the distances of the output vectors over the set of training output vectors. It is expected that this approach obtains more information than the centroid approach due to the use of all the samples.

In the incremental step, when instances of the new class are fed into the network (without retraining the network) the outputs obtained are added to the search space with their corresponding label and the test is performed using the kNN method.

The main disadvantage of this proposal is that the features of every instance of the training data need to be stored for classification. This is a disadvantage compared with the previous approach, in which only one element for each class is required, that is, the mean centroid.

5.1.3 Incremental single network results

The following results were obtained for two datasets. For the [Torres-García et al., 2016] dataset, the new class accuracy increased faster when the centroid approach was considered. The kNN method starts with high total accuracy; however, the incremental class does not increase rapidly when more instances were added, and the total accuracy shows a slight decrease, as shown in Fig. 5.4.

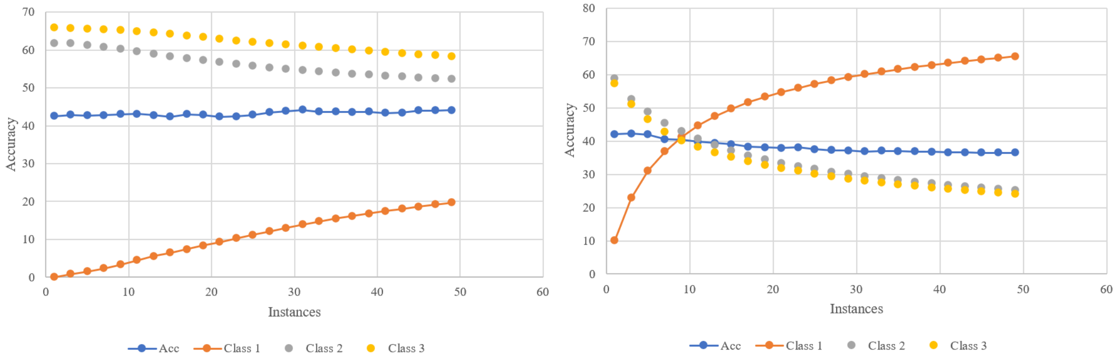


(a) Centroid-based incremental network

(b) kNN-based incremental network

Figure 5.4: Results for [Torres-García et al., 2016] dataset

For [Nguyen et al., 2017] dataset, the kNN approach exhibited a decrease in the performance of the original classes as more instances of the new class were added to the model. The centroid method exhibited a slow increase in new class accuracy, as shown in Fig. 5.5.



(a) Centroid-based incremental network

(b) kNN-based incremental network

Figure 5.5: Results for [Nguyen et al., 2017] dataset

The incremental results obtained using the kNN-based incremental network for both datasets presented lower accuracy than the centroid method.

In general, the behavior of this approach could indicate that using this architecture, the network outputs themselves are not well-distributed features. Thus, processing these outputs is required to increase pattern recognition. This assumption was explored in the following experiments.

5.2 Incremental twin network

Considering the results of previous approach, it was decided to improve the mean centroid network by implementing the following proposals. The inclusion of multiple class centroids, the use of independent networks for old and new classes, and classification based on multiple distance matrices.

First, the network is trained for the original classes. The outputs were used as feature vectors, which were converted into centroids using k-means clustering. Thus, the number of clusters per class can be greater than one, this can be seen as a general case of the previous network, as shown in Fig. 5.6.

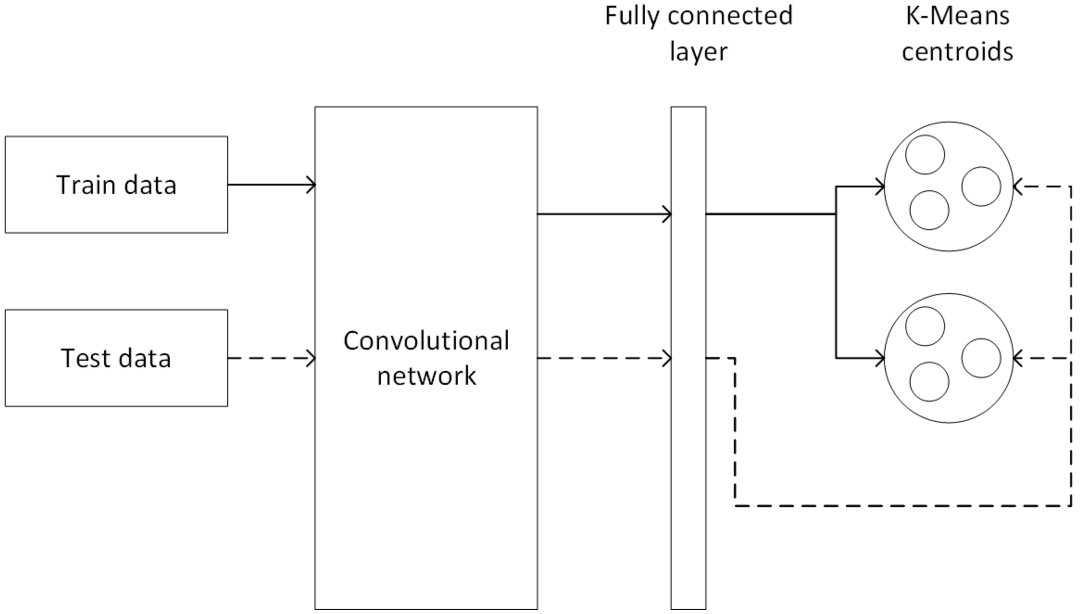


Figure 5.6: Non-incremental training and test step

The loss function is based on a standard function for clustering methods that aims to increase the distance of centroids of different classes and reduce the distance of centroids of the same class. A novel adaptation was implemented to consider that multiple centroids could be calculated for each class. The training instances were fed through the network to generate the centroids. Then, for each instance, the distance function d computes the closest centroid C to assign a class, as shown in Eq. 5.1.

$$L = -\sum_{i=1}^{n} log \frac{-e^{d(v_i, min(c_k))}}{\frac{1}{K}\sum_{m=1}^{K} -e^{d(v_i, c_m)}} \tag{5.1}$$

Where d is the distance function, n are the instances, K is the number of classes, C are the class centroids and V are the feature vectors of the instances.

For testing the performance, test data follows the same approach, it is fed through the network and labeled according to the closest centroid. Once the network was trained with the *original* classes, a *new* class could be added. To achieve this, a new network is created and trained without disturbing or retraining the old network, as shown in Fig. 5.7.

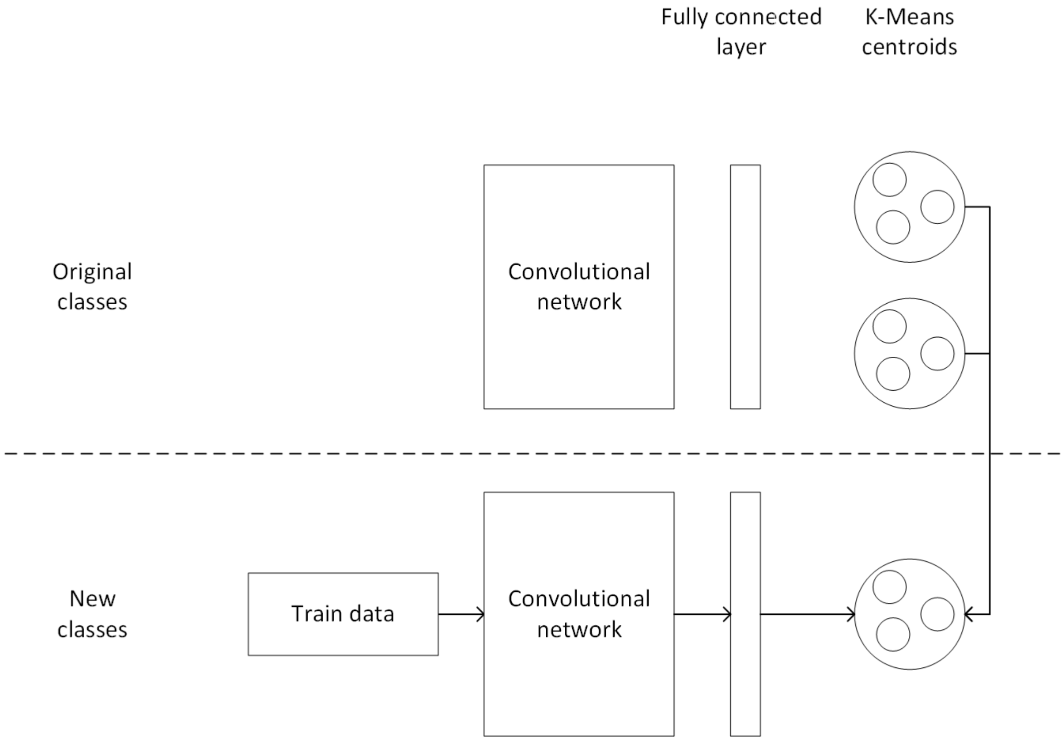


Figure 5.7: Incremental training step

The centroids of the old network are considered in the distance function to compute the loss function of the new network. This is similar to the *Knowledge Distillation*. However, in the proposed approach, the original network is similar in size to the new network.

Finally, to test the performance of the model with the new class, the test data (instances of both the original classes and the new class) are fed through both networks and the distances to the centroids are saved into a distance matrix (see Fig. 5.8).

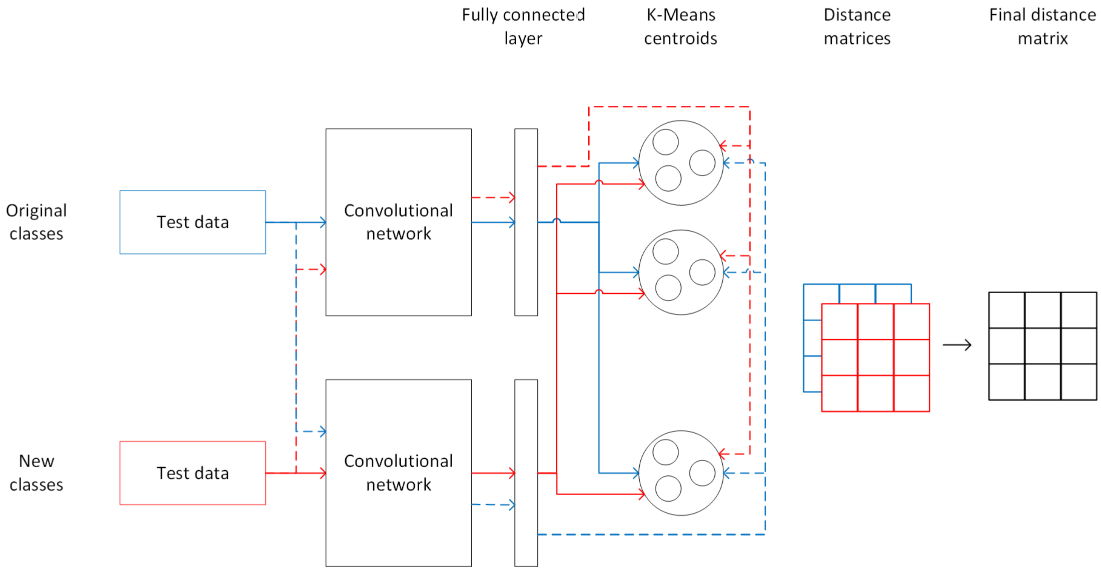


Figure 5.8: Incremental test step

Thus, two distance matrices are obtained, for the classification step, both matrices are compared and the lower values are preserved into a final distance matrix that will label test instances.

The complete method is presented in Algorithm 1. Where $O\_TrainData$ and $O\_TestData$ refer to the non-incremental dataset. $I\_TrainData$ and $I\_TestData$ are the incremental datasets. O_Labels and I_Labels denote the corresponding labels. C_O and C_I are the non-incremental and incremental classes respectively.

Algorithm 1 Incremental network

```
Input: $O\_TrainData$, $O\_TestData$, $I\_TrainData$, $I\_TestData$, $O\_Labels$, $I\_Labels$, $C_O$, $C_I$
Output: $y$
$Net_1 \leftarrow Initialize$                                  ▷ Netwwork Training
for $Epoch = 1$ To 50 do
    $X \leftarrow$ Feed $Net_1$ with $O\_TrainData$
    for all $Class$ In $C_O$ do
        $K \leftarrow$ kMeans($X[Class]$)                      ▷ Number of clusters in kMeans is set to 3
    end for
    for $Batch = 1$ To 40 do
        $X \leftarrow$ Feed $Net_1$ with $O\_TrainData[Batch]$
        $D \leftarrow distance(K, X)$                          ▷ See Algorithm 2
        $Loss \leftarrow \ell(D)$                              ▷ See Eq. 5.1
        $Net_1 \leftarrow Backpropagation(Net_1, Loss)$
    end for
end for
$X \leftarrow$ feed with $O\_TrainData$                        ▷ Network testing
$D \leftarrow distance(K, X)$                                  ▷ See Algorithm 2
$y \leftarrow evaluate(D, O\_Labels)$                          ▷ See Algorithm 3
$Net_2 \leftarrow Initialize$                                  ▷ Incremental Network Training
for $Epoch = 1$ To 100 do
    $X \leftarrow$ feed $Net_2$ with $I\_TrainData$
    for all $Class$ In $C_I$ do
        $K \leftarrow$ kMeans($X[Class]$)                      ▷ $K$ contains previous centroids and the new
    end for
    for $Batch = 1$ To 40 do
        $X \leftarrow$ feed $Net_2$ with $I\_TrainData[Batch]$
        $D \leftarrow distance(K, X)$                          ▷ See Algorithm 2
        $Loss \leftarrow \ell(D)$                              ▷ See Eq. 5.1
        $Net_2 \leftarrow Backpropagation(Net_2, Loss)$
    end for
end for
$X_1 \leftarrow$ feed $Net_1$ with $O\_TestData$               ▷ Incremental Network Testing
$X_2 \leftarrow$ feed $Net_1$ with $I\_TestData$
$D_1 \leftarrow distance(K, [X_1, X_2])$                       ▷ See Algorithm 2
$X_1 \leftarrow$ feed $Net_2$ with $O\_TestData$
$X_2 \leftarrow$ feed $Net_2$ with $I\_TestData$
$D_2 \leftarrow distance(K, [X_1, X_2])$                       ▷ See Algorithm 2
for all $Column$ In ($D_1$ Or $D_2$) do
    for $Row = 1$ To $lenght(K)$ do
        $D_f \leftarrow min(D_1[Column, Row], D_2[Column, Row])$
    end for
end for
$y \leftarrow evaluate(D_f, [O\_Labels, I\_Labels])$           ▷ See Algorithm 3
```

Algorithm 2 presents the distance function used to generate distance matrices. Where X are the network outputs and K is the set of k-means centroids of all classes. The cosine distance was chosen because of the dimensionality of the data.

Algorithm 2 function distance(K,X)

```
Input: $K, X$
Output: $D$
for all $Element$ In $X$ do
    for all $Centroid$ In $K$ do
        $D[Element, Centroid] \leftarrow cosine\_distance(Element, Centroid)$
    end for
end for
```

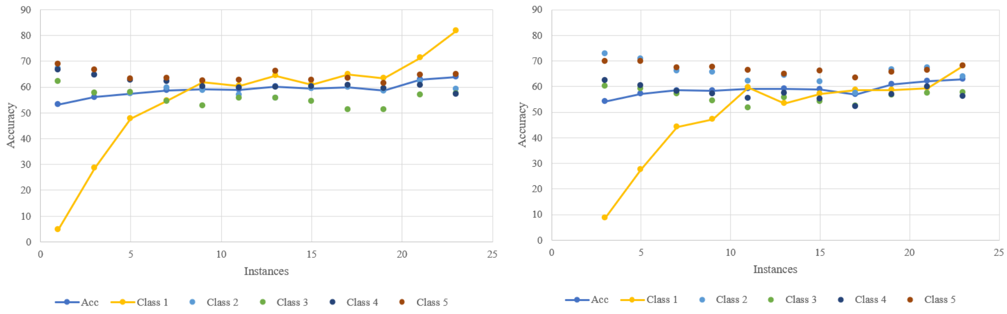
Algorithm 3 was developed to evaluate the calculated distances. Where *D* is a distance matrix in which the columns contain the instances of the test data, the rows contain the centroids, and *Labels* denotes the true class of the instances.

Algorithm 3 function evaluate(D,Labels)

```
Input: $D$, $Labels$
Output: $y$
for all $Column$ In $D$ do
    $X \leftarrow min(D[Column, :])$
    if $X = Label[Column, :]$ then
        $y \leftarrow y + 1$
    end if
end for
$y \leftarrow y/length(Labels)$
```

5.2.1 Incremental twin network results

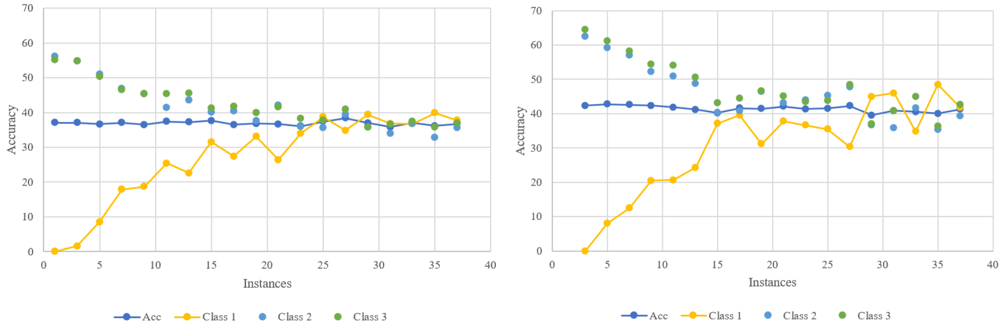
Using the method described in the previous section, the following results were obtained. In Fig. 5.9, 5.10 and 5.11 the results for the three datasets are presented. As mentioned before, the results for each dataset are the average from 6 classifiers, one for each subject. In addition, the experiment were repeated 5 times for each subject. The figures show the behavior of all the classes, the incremental class, i.e., Class 1, highlighted in continuous yellow and the total accuracy in continuous blue. Instances of the new class were progressively added to the model to observe the trade-off between the number of instances and the accuracy.



(a) 1 centroid

(b) 3 centroids

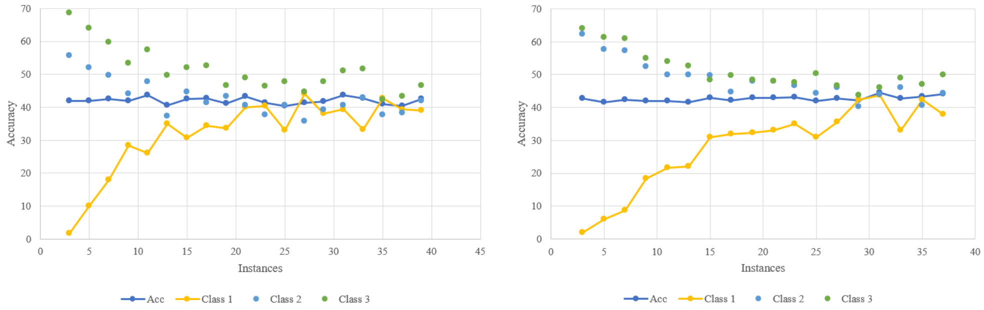
Figure 5.9: Incremental twin network results in [Torres-García et al., 2016] dataset, color blue line represents the total accuracy, color yellow line represents the incremental class accuracy



(a) 1 centroid

(b) 3 centroids

Figure 5.10: Incremental twin network results in [Nguyen et al., 2017] dataset, color blue line represents the total accuracy, color yellow line represents the incremental class accuracy



(a) 1 centroid

(b) 3 centroids

Figure 5.11: Incremental twin network results in new dataset, color blue line represents the total accuracy, color yellow line represents the incremental class accuracy

From these results, it can be observed that the change in the number of centroids has an impact on the total accuracy as well as on incremental class accuracy. In some cases the use of 3 centroids increased the total accuracy. In addition, the use of 1 centroid improved the incremental learning by obtaining higher accuracies using few classes. This trade-off of total accuracy and a faster incremental learning may rely in the differences of the dataset, i.e., channels number, acquisition protocols, sample rate.

5.3 Comparative results of the proposed approaches

Table 5.2 compares the results obtained with the two methods used in the single incremental network approach: kNN-based method and mean centroid method. For this purpose, the average accuracy for each increment in the new class was calculated. It was divided to show the total average accuracy and the incremental class average accuracy for each method. The total average accuracy results obtained for the [Torres-García et al., 2016] dataset using the centroid method were 61.86 ± 3.36 and for the kNN method, it was 59.14 ± 0.84. There was no statistical difference, both methods have a similar behavior, i.e., there was no significant loss in total accuracy when a new class was added. For the [Nguyen et al., 2017] dataset, the total accuracy was 43.56 ± 0.67 for the centroid method and 37.68 ± 1.7 for the kNN method. In this case, the first method obtained the best total accuracy, whereas the second method showed an increase in incremental accuracy and the old classes showed a decrease, indicating catastrophic forgetting.

For both datasets an ANOVA analysis showed a significant difference: $p = 0.0093$ for [Torres-García et al., 2016] and $p = 7.5163e^{-33}$ for [Nguyen et al., 2017]. For incremental learning the analysis showed $p = 0.3451$ for [Torres-García et al., 2016] and $p = 4.9588e^{-28}$ for [Nguyen et al., 2017].

In the [Torres-García et al., 2016] dataset, the centroid method exhibited a faster increase of accuracy in the new class than the kNN method (see Fig. 5.9). More importantly, there was no decrease in the original classes while more instances of the new class were added. For the [Nguyen et al., 2017] dataset, the centroid method exhibited a slower increase than the kNN method. Nevertheless, the original classes showed a rapid decrease with the kNN method (see Fig. 5.10).

Table 5.2: Incremental single network total accuracy results.

Dataset	**Centroid method**		**kNN method**	
	Total	Incremental	Total	Incremental
[Torres-García et al., 2016]	61.86 ± 3.36	40.23 ± 18.06	59.14 ± 0.84	34.66 ± 10.43
[Nguyen et al., 2017]	43.56 ± 0.67	15.39 ± 8.02	37.68 ± 1.7	58.4 ± 13.79

The next approach aims to improve the results obtained by using a centroid based incremental network. The proposed improvements were the use of multiple centroids, independent networks for the incremental approach and the adaptation of distance computing to the centroids. As shown in Table 5.3, the [Torres-García et al., 2016] dataset obtained an accuracy of 58.89 ± 2.4 using one centroid and 59.02 ± 2.7 using three centroids, and there was no statistical difference in this case. The accuracies obtained in the [Nguyen et al., 2017] dataset were 41.43 ± 0.9 and 36.91 ± 0.59 for three and one centroids, respectively, indicating a better performance using three centroids. Finally, for the new dataset, an accuracy of 42.64 ± 0.81 was obtained for three centroids, and 41.97 ± 1 for one centroid, there was no statistical difference.

ANOVA for total accuracy obtained: $p = 0.5221$ for [Torres-García et al., 2016] dataset, $p = 0.12103e^{-18}$ for [Nguyen et al., 2017] dataset and $p = 0.0363$ for the new dataset. And respectively, the test for incremental class accuracy obtained: $p = 12.09$, $p = 0.6041$ and $p = 0.1432$. The incremental learning accuracies of the three datasets did not show statistical differences, but some cases showed a faster increase using fewer instances, e.g., according to Fig. 5.11 for one centroid fewer instances of the new class were needed to reach the total accuracy in the new dataset.

Table 5.3: Incremental twin network total accuracy results.

Dataset	1 centroid		3 centroids	
	Total	Incremental	Total	Incremental
[Torres-García et al., 2016]	59.02 ± 2.78	55.38 ± 20.49	58.89 ± 2.4	49.29 ± 17.21
[Nguyen et al., 2017]	36.91 ± 0.59	26.91 ± 12.52	41.43 ± 0.9	30.57 ± 13.63
New dataset	42.64 ± 0.81	31.92 ± 11.19	41.97 ± 1	28.22 ± 12.53

Because the centroid based network showed a better performance than the kNN based network, it is expected that the proposed incremental twin network will improve the performance of the method. This was confirmed because the incremental twin network showed high total accuracy and a faster increase in incremental class accuracy. The increase in the incremental class accuracy using a few instances is advantageous for BCIs because it can reduce the time required for training.

5.4 Comparative results with previous work

Although there are many incremental learning approaches for image classification, they are usually oriented towards increasing the number of classes with a large number of instances. Therefore, proposed method was compared with a similar method which was applied to a different task. This provides experimental evidence of the competitive performance of incremental learning when applied to small dataset scenarios.

The obtained results were compared with the proposal of [Cheng et al., 2019], where a similar incremental approach was presented for odor classification. Despite the use of a different task, the method is oriented to a scenario similar to that proposed in this study, in which few classes are added to a neural network approach.

The results for [Torres-García et al., 2016] dataset are shown in Fig. 5.12.

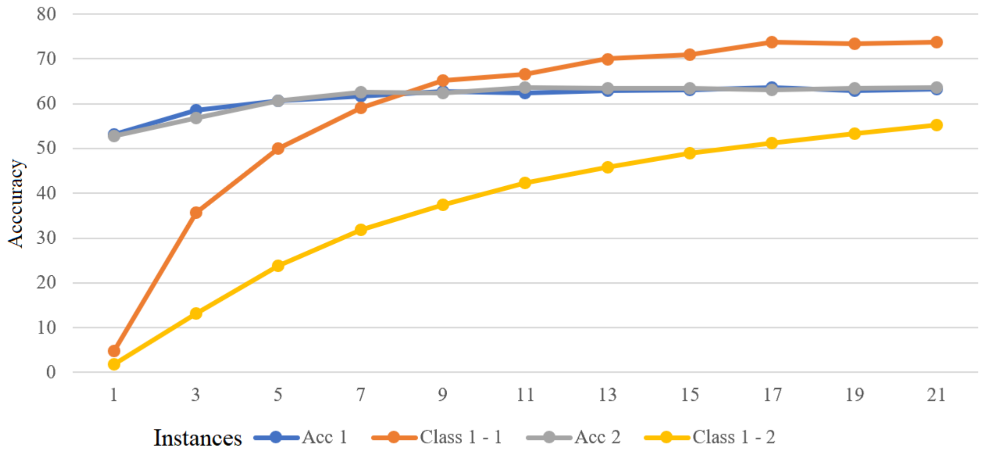


Figure 5.12: [Torres-García et al., 2016] dataset. The results of the proposed method are in color blue for the total accuracy and the incremental class accuracy in color orange. For the [Cheng et al., 2019] method, the total accuracy is presented in color gray and the incremental class accuracy in color yellow.

The results for the [Nguyen et al., 2017] dataset are shown in Fig. 5.13.

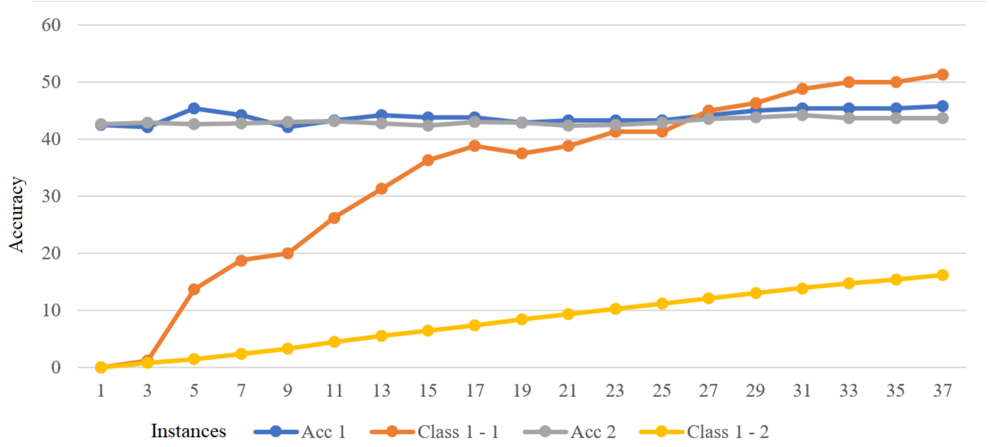


Figure 5.13: [Nguyen et al., 2017] dataset. The results of the proposed method are in color blue for the total accuracy and the incremental class accuracy in color orange. For the [Cheng et al., 2019] method, the total accuracy is presented in color gray and the incremental class accuracy in color yellow.

The results for the new dataset are presented in Fig. 5.14.

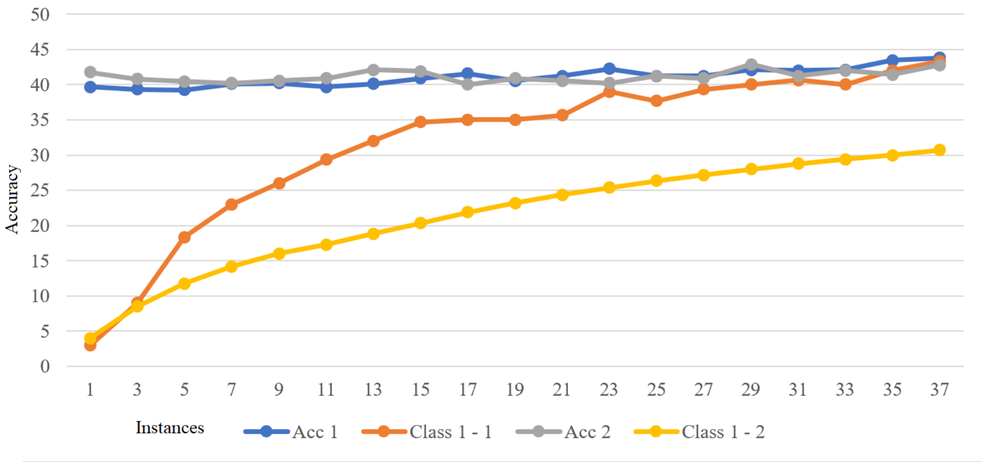


Figure 5.14: New dataset. The results of the proposed method are in color blue for the total accuracy and the incremental class accuracy in color orange. For the [Cheng et al., 2019] method, the total accuracy is presented in color gray and the incremental class accuracy in color yellow.

The comparative results with the [Cheng et al., 2019] approach show that both methods have a similar stable total accuracy performance for all datasets. Nevertheless, the incremental accuracy results for class 1 showed a notable difference, see Table 5.4. In all the cases the incremental accuracy is higher using the proposed method, and it increased using fewer instances.

ANOVA for total accuracy obtained: $p = 0.9530$ for [Torres-García et al., 2016] dataset, $p = 0.0043$ for [Nguyen et al., 2017] dataset and $p = 0.7582$ for the new dataset. Respectively, the test for incremental class accuracy showed: $p = 0.0047$, $p = 1.2958e^{-07}$ and $p = 0.0021$.

Table 5.4: Comparative total accuracy results.

Dataset	**Proposed method**		**Baseline method**	
	Total	Incremental	Total	Incremental
[Torres-García et al., 2016]	61.7 ± 3.16	58.84 ± 20.91	61.45 ± 3.51	36.8 ± 17.5
[Nguyen et al., 2017]	43.96 ± 1.18	33.48 ± 15.97	43.06 ± 0.53	8.23 ± 5.23
New dataset	41.08 ± 1.3	31.73 ± 11.26	41.19 ± 0.84	21.37 ± 7.73

Both approaches maintain a total accuracy that does not decay for any number of instances of the new class, i.e., there is no catastrophic forgetting. Nevertheless, the comparison showed two improvements in the proposed incremental approach concerning the previous work. The incremental class achieved a stable accuracy with fewer instances. In addition, higher accuracy was achieved for the incremental class. These improvements provide the method with a faster adaptation of the new class using fewer instances while maintaining the total accuracy.

Chapter 6

Conclusions and future work

The present study aimed to recognize imagined speech using EEG and increase the vocabulary of the recognized words. To achieve this, different methods have been proposed to find a good compromise between classification and incremental learning of imagined speech. In Chapter 4, the recognition of imagined speech is analyzed and two methods are proposed: dictionary learning and neural networks. The dictionary approach provides a better interpretation of the signals; however, feature extraction must be defined precisely. The advantages of neural networks include automatic feature extraction and flexibility in modifying their architecture for different purposes. However, these methods lack interpretability. Furthermore, in Chapter 5, incremental learning is analyzed with the objective of adding a new imagined word to an already trained model. Two approaches were proposed: incremental single networks and incremental twin networks. A single network aims to add a new word without modifying the network architecture, only the network outputs. Twin networks extend network architecture to include new classes.

6.1 Conclusions

For imagined speech recognition, the dictionary approach achieved an accuracy similar to that of neural networks; however, signal processing and feature extraction must be defined manually. Neural networks include automatic feature extraction, which aids in the identification of adaptable models for various subjects and datasets. Another consideration when choosing neural networks is their flexibility in implementing incremental learning, which was an objective of this study. Nevertheless, a major issue is catastrophic forgetting, which is common in these approaches and appears in some of the proposed architectures.

Furthermore, two incremental learning approaches have been proposed: an incremental single network and an incremental twin network. The latter presented better results because of the proposed modifications: model extension for multiple centroids, complementary networks for incremental classes, loss function adaptation, and evaluation based on multiple distance matrices.

The results of incremental twin networks showed a stable total accuracy, and there was no drop that resembled *catastrophic forgetting*. For the [Torres-García et al., 2016] dataset, a good relationship exists between the *stability* and *plasticity* for every subject. For the datasets in [Nguyen et al., 2017] and the new dataset, some subjects exhibited a decrease in accuracy for the old classes, which tended to recover when more instances of the new class were added. Nevertheless, this decrease did not recover for a few subjects (see Appendix B).

An explanation for the variations in the behavior of the datasets is that they differ in the number of channels; [Torres-García et al., 2016] has 14 channels, [Nguyen et al., 2017] has 60 channels, and the new dataset has 31 channels. A small number of channels may allow a better fit of the network with few parameters. Another consideration is the number of classes; [Torres-García et al., 2016] includes five classes, [Nguyen et al., 2017] and the new dataset has only three classes. A higher number of old classes may allow robustness of the model that is not perturbed by the inclusion of a new class. Another consideration is that the acquisition protocol was different for each dataset; [Torres-García et al., 2013] allowed the subjects to manually determine when they had finished imagining the words, [Nguyen et al., 2017] fixed a period in which the subjects could repeat the imagination of the word several times, and in the new dataset a fixed time was proposed for the subjects to repeat the word once.

As a final remark, the proposed method achieved good performance in contrast to other incremental class learning tasks, in particular, incremental learning for images. A review of various incremental learning methods was presented in [Tao et al., 2020, Masana et al., 2020], most of which focus on mitigating catastrophic forgetting. These studies showed that the total accuracy decays as more instances of a new class are added, which did not occur in the proposed method. It is important to emphasize that these are different tasks with significant differences. Moreover, these studies used large image datasets such as CIFAR100, which contains a large number of classes and instances.

The incremental twin network has a better performance for incremental learning than the method proposed in [Cheng et al., 2019]. The results showed a faster increase in the accuracy using less instances. This behaviour is desirable for BCI due to the reduction of time in the data acquisition.

6.2 Future work

In this section, ideas for extending the proposed method are presented. An open problem in brain signal analysis is the variation in the signals for different subjects. Therefore, the data available to create a model depends on the data of one subject. The possibility of using data from multiple users may increase the robustness of models and allow their extension to new subjects. Moreover, creating a specific model for each subject is a costly but viable way to increase the recognition rates of models. This implies, for example, changing the parameters of the networks, feature extraction process, or number of centroids.

Improvements in signal processing can be applied to channel selection. In certain cases, the use of several channels increases the probability of adding noise or deficient sensors. Therefore, removing these imperfections could improve the performance of these methods. This concept can be extended to frequency transformations and the selection of frequency bands that provide the relevant task information.

The amount of available data may be a significant factor in network performance. To analyze this behavior, either one of the databases can be augmented, or the other can be reduced. Data augmentation can be performed by adding data from different subjects. Moreover, the use of overt speech EEG signals to improve imagined speech recognition can be explored, which can improve the generation of models and can be extended for handicapped persons to whom training a model is complicated.

Finally, to implement a complete BCI which include the proposed method will allow to identify flaws in the model and to propose improvements.

Appendix A

Incremental dictionary approach

In this appendix, an incremental learning experiment using dictionaries is presented. The Fig. A.1 represents the general method flowchart, which is based on the work presented in [Wang et al., 2013b]. This method was applied to all the participants.

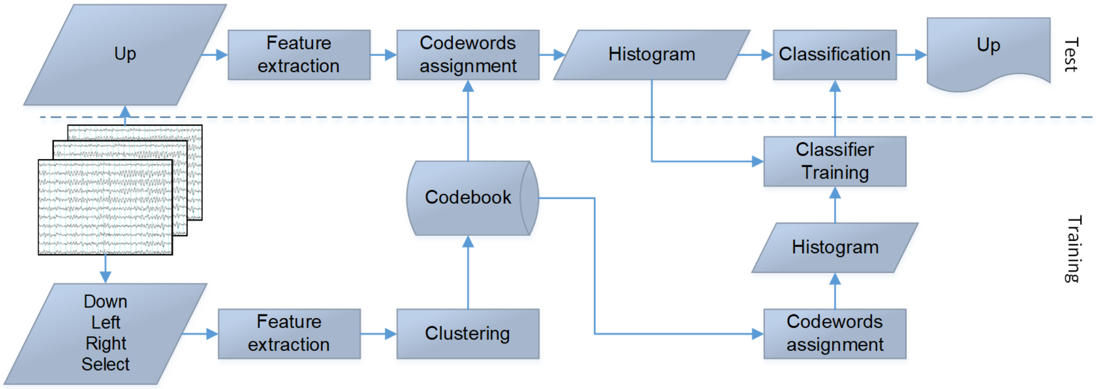


Figure A.1: General method flowchart

The previously extracted features were used to generate a codebook using a clustering method, i.e. k-means, to obtain the most representative features from the data. As in [Plinge et al., 2014], the clustering method was applied to each class. Then, for each class, *k* clusters were obtained and named codewords. Subsequently, the resulting *k* prototypes were concatenated into a single codebook [Lazebnik and Raginsky, 2009].

A drawback of using k-means is that the number of clusters must be defined *a priori*. Knowledge of the problem and the data is essential for defining the number of clusters. Due to the few reported studies on imagined speech analysis, this issue was overcome by applying a genetic algorithm to determine an appropriate number of clusters based on classification accuracy [García-Salinas et al., 2017]. Clusters were obtained by considering the classification accuracy as the objective function. Therefore, the mean accuracy for each subject was used to obtain a unique cluster number. Such experiments obtained a cluster number k of 250.

These numbers will be fixed for the following experiments. Once the codebook was generated, the next step was to replace every instance in matrix y that generates the codebook with one of the k codewords. The result was a sequence of codewords over the original data. To choose the codeword that replaced each instance, a similarity measure was applied, i.e., Euclidean distance.

At this step, the original signals became sequences of codewords. The occurrence of codewords was then counted in each word epoch, for which a convenient representation was a histogram. The results were a set of histograms, each representing an epoch of differently imagined words from each subject.

Once the signals were transformed into a set of histograms, the classes, i.e. imagined words, were associated with the corresponding set of histograms which represented them. Thus, each histogram was considered a classification instance. To analyze these histograms, a Naive Bayes classifier was applied. For this purpose, 75% of the word epochs were used to generate the codebook and train the classifier, the remaining 25% were reserved for testing. Moreover, owing to the random properties of k-means clustering, this method was applied to a 10-cross-fold validation of the partitions.

To test the transfer learning, codebook generation was performed using only four words from the dataset. Subsequently, instances of the new imagined word were replaced using a codebook generated with these previous words. Thus, a set of histograms was generated from the new word, which was associated with this new class, and finally merged with the previous class histograms for classifier training.

This set of histograms was used to find patterns able to discriminate the new class using codewords generated by different imagined words. the aim was to verify whether a new imagined word could be represented and discriminated using a previously generated codebook.

The objective of the calibration was to observe the classification performance by adding different amounts of instances of the new class in the codebook generation. A model that can discriminate between classes without using the data from a new class is preferable. Nevertheless, in some cases, small amounts of data are required for model generation in order to improve the discrimination of a new class.

A.1 Results

First, the baseline result of the method was obtained using the entire vocabulary, i.e. five words were used in codebook generation, achieving an average accuracy of 65.65 ± 13.39. In Fig. A.2 a comparison of the proposed representation and [Torres-García et al., 2016] is presented.

The average accuracy when transfer learning was applied to the “up” word was 58.74 ± 13.39. Moreover, when transfer learning is applied to the “down” word, the average accuracy was 61.38 ± 12.47. It should be noted that transfer learning results were obtained without calibration.

To compare the transfer learning behaviors, a baseline confusion matrix is presented in Table A.1. The following confusion matrices were generated by averaging the confusion matrices from all subjects, and are presented as global percentages for easy interpre-

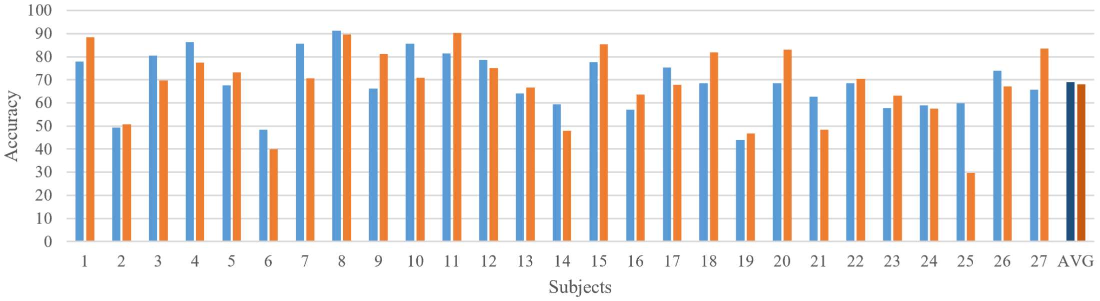


Figure A.2: Accuracy results comparison without transfer learning (proposed method in blue, [Torres-García et al., 2016] in orange)

tation.

Table A.1: Baseline confusion matrix.

	Up	Down	Left	Right	Select
Up	75.93	9.26	3.21	6.42	5.19
Down	8.89	61.48	7.78	14.57	7.28
Left	2.84	6.91	60.00	15.31	14.94
Right	3.83	12.96	13.33	61.36	8.52
Select	3.83	7.04	13.95	5.68	69.51

The confusion matrix of ”up” word transferring is presented in Table A.2.

Also, in Fig. A.3 the results per subject for the class ”Up” transferring are presented.

In addition, Table A.3 corresponds to the “down” word transferring classification matrix, which shows a different behavior from that of the transferred class. The accuracy obtained from “up” word decreases 8.52 in comparison to the baseline. Otherwise, the “down” word accuracy increased in 2.35.

In addition, the average histogram from all the subjects was obtained to calculate the codewords used by the new class. This analysis should complement the confusion matrix analysis by comparing the confusion among classes and the percentage of codewords

Table A.2: "Up" class transferring confusion matrix.

	Up	Down	Left	Right	Select
Up	67.41	12.10	6.79	9.01	4.69
Down	15.19	56.17	8.52	13.21	6.91
Left	10.00	14.20	50.12	12.59	13.09
Right	10.49	20.37	10.62	51.23	7.28
Select	8.40	10.00	8.52	4.32	68.77

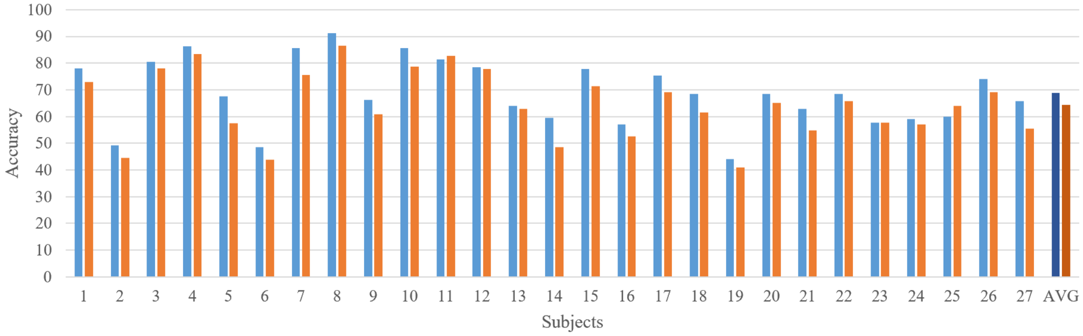


Figure A.3: Transfer learning results for the word "up" (baseline in blue, transfer results in orange)

used. Table A.4 summarizes the codeword percentages used in the transferred classes by averaging the results from all the subjects.

To improve the classification results, a small number of epochs of new words were used for codebook generation. Subsequently, the same number of instances used in codebook generation was used for classifier training. The class imbalance presented in this calibration was not considered.

Table A.5 shows the accuracies of the transfer learning approach using different amounts of epochs for the "up" word transfer.

Table A.6 shows the accuracies from the "down" word in a transfer learning approach using different amounts of its epochs in the codebook generation step.

The last two tables show the accuracy of the transferred classes and the total accu-

Table A.3: “Down” class transferring confusion matrix.

	Up	Down	Left	Right	Select
Up	70.99	14.69	6.17	4.94	3.21
Down	9.88	58.52	13.09	13.58	4.94
Left	3.09	11.36	58.40	13.21	13.95
Right	3.70	18.27	20.00	52.59	5.43
Select	2.72	8.27	19.14	3.46	66.42

Table A.4: Codewords distribution percentages to represent the transferred classes.

	Up	Down	Left	Right	Select
Up	-	22.45	23.26	26.02	27.56
Down	18.56	-	24.18	27.23	29.31

racy of the five classes obtained by averaging the results for all subjects.

A.2 Discussion

The baseline accuracy when no transfer learning was applied was 65.65 ± 13.39, which is not significant compared with the baseline work in [Torres-García et al., 2016] with $[F(1,52) = 0.4, p = 0.5323]$, according to a one-way analysis of variance of the subjects. When applying transfer learning, a mean accuracy of 58.74 ± 13.39 was obtained for the “up” word, which is an accuracy reduction of 6.91. When the “down” word is transferred a mean accuracy of 61.38 ± 12.47 was achieved, this is an accuracy decrease of 4.27. The transfer showed a slight decrease compared with the baseline accuracy. Moreover, a one-way analysis of variance (ANOVA) of the obtained results was conducted to compare the effects of the transfer learning approach for the two words and the baseline method. There was no significant effect $[F(1,52) = 3.6, p = 0.0634]$ for the ”up” word and $[F(1,52) = 1.47, p = 0.2305]$ for the ”down” word.

Table A.5: “Up” class accuracies using epochs of this class in the codebook generation.

Epochs number	“Up” accuracy	Total accuracy
0	67.41 ± 19.22	58.74 ± 13.39
1	19.75 ± 18.00	56.77 ± 11.70
3	30.99 ± 19.59	61.83 ± 10.96
5	39.14 ± 21.83	61.60 ± 12.78
8	47.04 ± 24.01	62.79 ± 13.03
10	51.36 ± 23.47	62.79 ± 13.03

Table A.6: “Down” class accuracies using epochs of this class in the codebook generation.

Epochs number	“Down” accuracy	Total accuracy
0	58.52 ± 24.62	61.38 ± 12.47
1	15.19 ± 13.34	58.07 ± 11.08
3	22.96 ± 16.44	60.59 ± 10.95
5	27.90 ± 16.70	61.83 ± 10.96
8	33.70 ± 22.91	62.59 ± 12.92
10	38.52 ± 22.73	62.74 ± 13.06

In Table A.2, the “up” word transferring confusion matrix shows confusion between this word and the “down” word. Moreover, as expected, most of the words increased their confusion with the word ”up” due to the results in Table A.4 which showed that this new word is represented by instances of every word.

The “down” word shows a different behavior, results in Table A.3, showed an accuracy of 61.38 ± 12.47. the results did not correspond to the codeword distributions in Table A.4. The “down” word codification included more codewords of the “select” word. Thus, a higher confusion between the words ”down” and ”select” was expected. Nevertheless, the ”down” transferring confusion matrix shows a higher confusion with the “right” word.

In Table A.5, the accuracy when the calibration data is added to the “up” word transferring is presented. As expected, accuracy increased when more calibration instances were included. Nevertheless, better performance was achieved when there was no information about the class. Otherwise, Table A.6 shows the accuracy when the ”down” word was transferred. It can be noticed that the transferring of this word shows a similar behavior as the ”up” word.

The performance decrement when calibration is applied can be attributed to three causes: (1) the use of few data to represent the transferred class, (2) the use of non-representative data of the new class, and (3) class imbalance during classifier training. In practical applications, calibration instances can be considered instances of a new word that a user wants to learn. Through a deeper analysis, these instances can be exploited to improve the knowledge of new imagined words and increase recognition using a few calibration instances.

A.3 Conclusions

The proposed method is capable of extending an imagined speech vocabulary with a slight decrease in accuracy for a set of subjects. By excluding the data of the transferred words from the generation step, similar accuracy results were obtained as if the data were included. Nevertheless, the use of few calibration data did not increase the classification accuracy. In practical applications, this calibration step requires less information possible from the transferred words. It must be considered that, if the calibration data are not representative of the imagined word, the codebook will not represent correctly such word and the classification performance will decay.

It is possible to add more than one word to the model. Nevertheless, it is expected that the classification performance will decrease. Further experiments may analyze the behavior of the method when more words are added, considering different combinations of the added words and calibration steps.

The analysis of the codewords distribution showed that the number of codewords used from other imagined words is not correlated to the confusion among classes. further analysis is required to confirm these results properly.

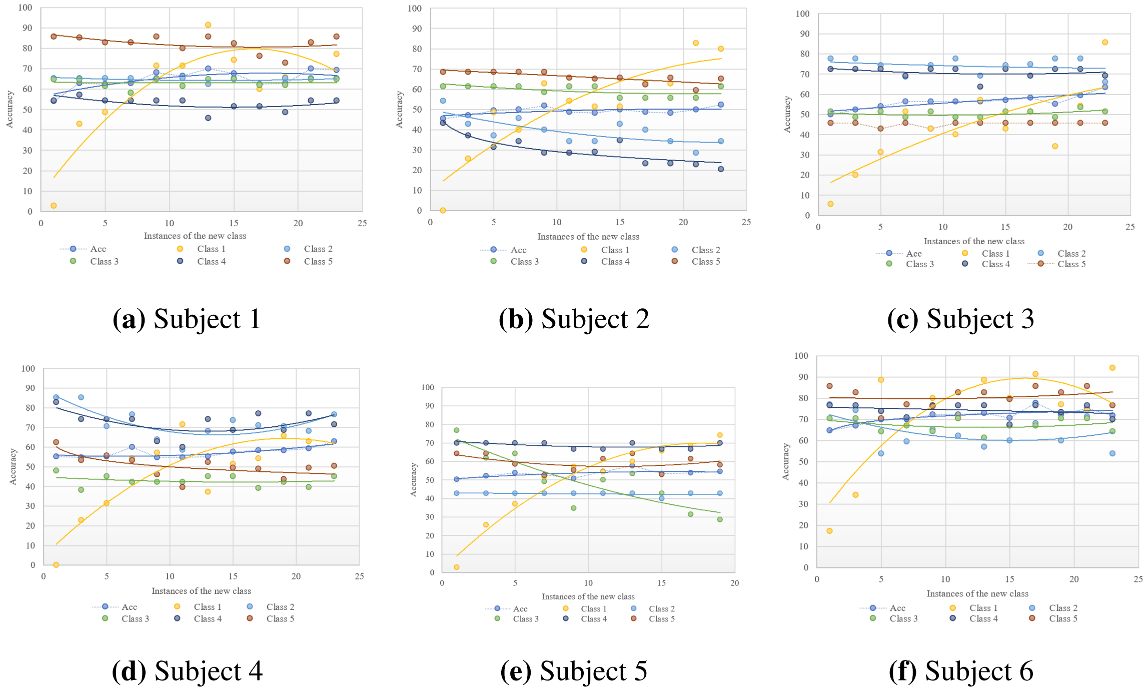
It is also interesting to highlight that the feature extraction step takes only into account the signal micro-volt values. Hence, the impact of noise on filtered signals must be explored. Additionally, in future experiments, the frequency information of the signal could be considered to search for patterns related to the brain activity frequency bands. Additionally, the omission of the frequency feature extraction step of the signal renders this method more suitable for real-time BCIs.

Appendix B

Proposed method results per subject

For a deeper analysis, the results per subject are shown below. The yellow line represents the new class accuracy, the medium blue the total accuracy, and the resting lines are the accuracy of the old classes. The lines also represent a polynomial tendency that models the behavior of each class while more instances of the new class are added. This results were obtained applying the kMeans proposed network using one and three centroids.

The aim of this analysis is to deepen in the previous results which are an average from all the subjects. It is expected that subjects show different behaviors. The brain has unique configurations for each person, which impacts the results of a general model for EEGs. Thus, some of the subjects may present different performances, and this results could aid to decide whether a method is suitable for some subjects under certain parameters.



(a) Subject 1

(b) Subject 2

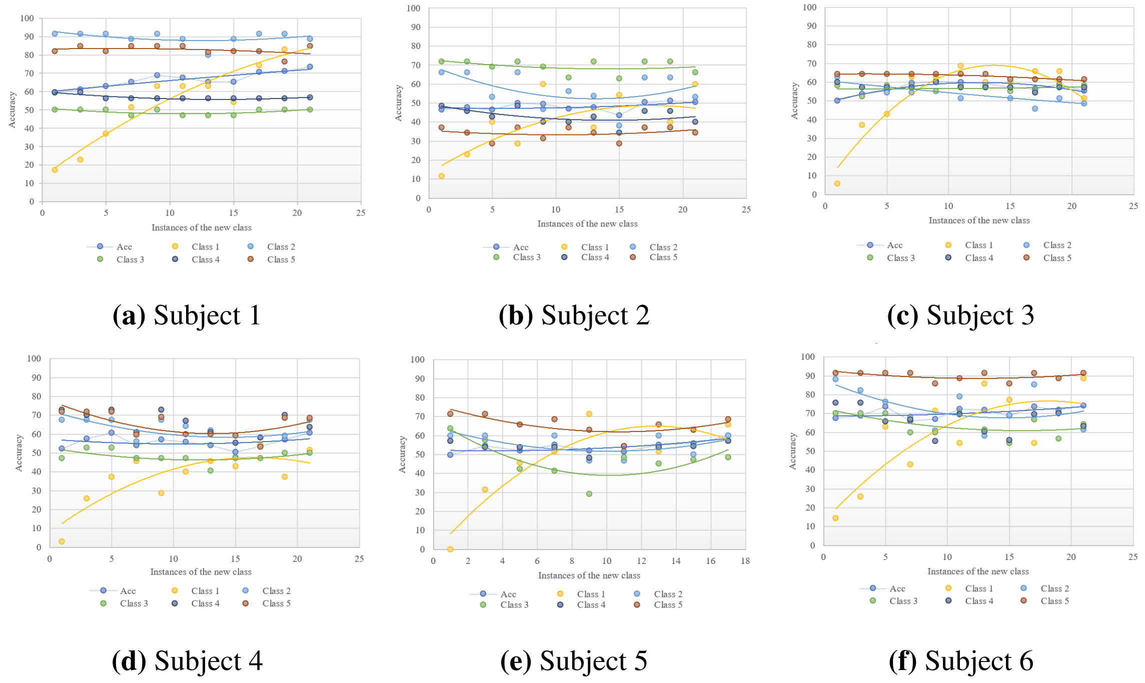
(c) Subject 3

(d) Subject 4

(e) Subject 5

(f) Subject 6

Figure B.1: [Torres-García et al., 2016] dataset, results per subject using 1 centroid



(a) Subject 1

(b) Subject 2

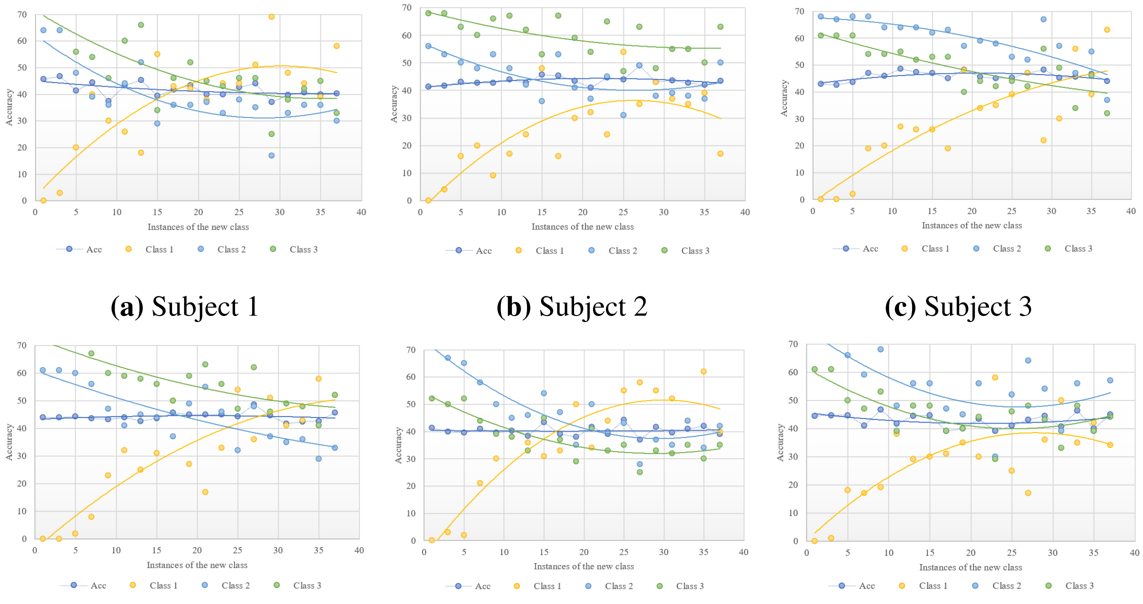
(c) Subject 3

(d) Subject 4

(e) Subject 5

(f) Subject 6

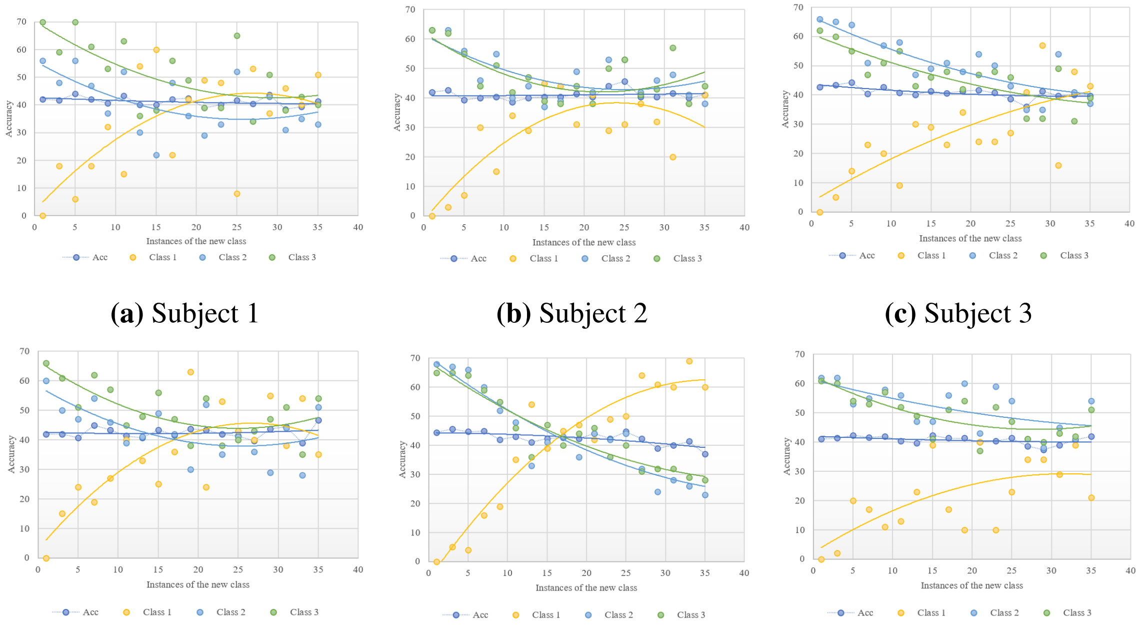
Figure B.2: [Torres-García et al., 2016] dataset, results per subject using 3 centroids



(a) Subject 1 **(b)** Subject 2 **(c)** Subject 3

(d) Subject 4 **(e)** Subject 5 **(f)** Subject 6

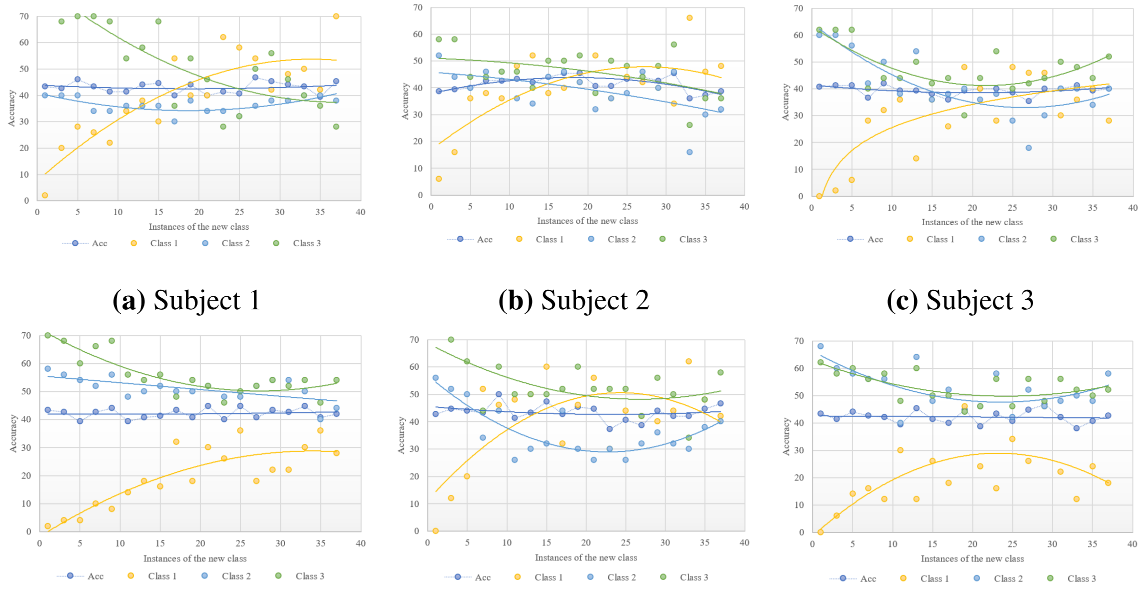
Figure B.3: [Nguyen et al., 2017] dataset, results per subject using 1 centroid



(a) Subject 1 **(b)** Subject 2 **(c)** Subject 3

(d) Subject 4 **(e)** Subject 5 **(f)** Subject 6

Figure B.4: [Nguyen et al., 2017] dataset, results per subject using 3 centroids



(**a**) Subject 1

(**b**) Subject 2

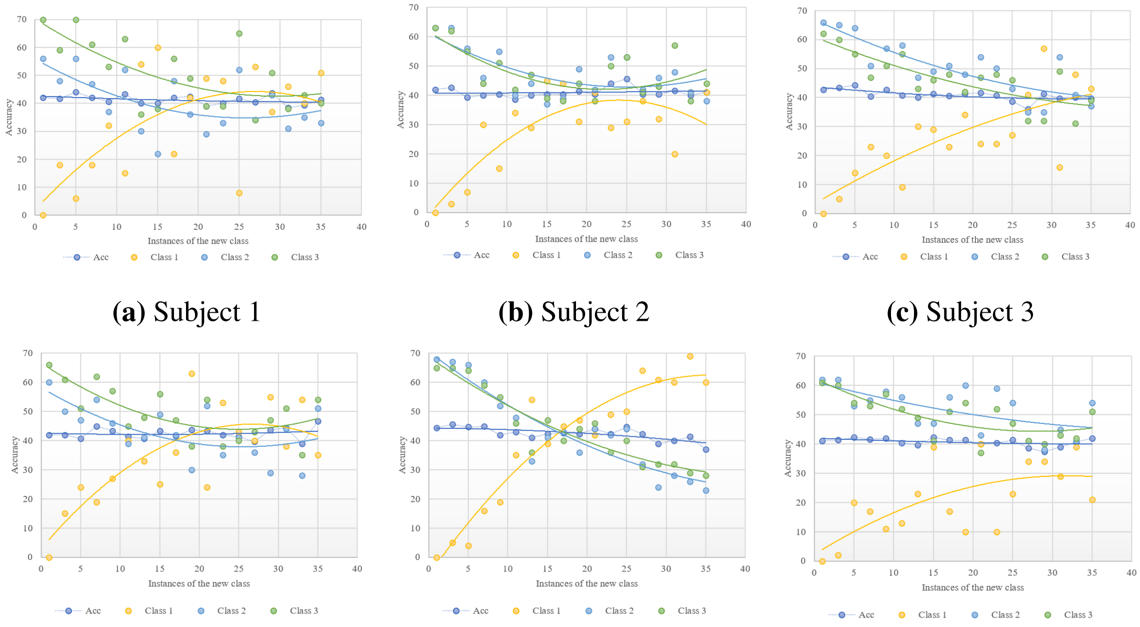
(**c**) Subject 3

(**d**) Subject 4

(**e**) Subject 5

(**f**) Subject 6

Figure B.5: New dataset, results per subject using 1 centroid



(**a**) Subject 1

(**b**) Subject 2

(**c**) Subject 3

(**d**) Subject 4

(**e**) Subject 5

(**f**) Subject 6

Figure B.6: New dataset, results per subject using 3 centroids

An analysis per subject for [Torres-García et al., 2016] dataset presents a high accuracy as well as a fast adaptation of the new class to the model for both methods. Moreover, polynomial trend analysis showed that there is no degradation of the original classes for most of the subjects.

On the other hand, for [Nguyen et al., 2017] dataset, the analysis per subject indicates a similar accuracy behavior for both methods. Regarding the incremental accuracy, when one centroid is employed, there is a decrease in the original classes while the new class increases, for most of the subjects this decrease does not recover. The use of three centroids shows a better behavior for the incremental step due to the original classes tend to recover after a considerable decrease of accuracy.

Finally, for the new dataset, the behavior is the opposite. When one centroid is used, there is a decrease of the original classes and afterward, a trend of recovery appears. When three centroids are created, the original classes show a trend of decrease that does not recover and the incremental class increases faster.

Bibliography

[Ade and Deshmukh, 2013] Ade, R. R. and Deshmukh, P. R. (2013). Methods for Incremental Learning: A Survey. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 3(4):119–125.

[Aharon et al., 2006] Aharon, M., Elad, M., and Bruckstein, A. (2006). rm K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322.

[Aleman et al., 2005] Aleman, A., Formisano, E., Koppenhagen, H., Hagoort, P., De Haan, E. H. F., and Kahn, R. S. (2005). The functional neuroanatomy of metrical stress evaluation of perceived and imagined spoken words. *Cerebral Cortex*, 15(2):221–228.

[Ameri et al., 2015] Ameri, R., Pouyan, A., and Abolghasemi, V. (2015). EEG signal classification based on sparse representation in brain computer interface applications. In *2015 22nd Iranian Conference on Biomedical Engineering (ICBME)*, pages 21–24. IEEE.

[Ameri et al., 2016] Ameri, R., Pouyan, A., and Abolghasemi, V. (2016). Projective dictionary pair learning for EEG signal classification in brain computer interface applications. *Neurocomputing*, 218:382–389.

[An et al., 2014] An, X., Kuang, D., Guo, X., Zhao, Y., and He, L. (2014). A deep learning method for classification of eeg data based on motor imagery. In *Lecture*

Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8590 LNBI, pages 203–210. Springer, Cham.

[Bart and Ullman, 2005] Bart, E. and Ullman, S. (2005). Cross-generalization: learning novel classes from a single example by feature replacement. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 672–679 vol. 1.

[Barthélemy et al., 2013] Barthélemy, Q., Gouy-Pailler, C., Isaac, Y., Souloumiac, A., Larue, A., and Mars, J. I. (2013). Multivariate temporal dictionary learning for EEG. *Journal of Neuroscience Methods*, 215:19–28.

[Barzegaran et al., 2017] Barzegaran, E., Vildavski, V. Y., and Knyazeva, M. G. (2017). Fine structure of posterior alpha rhythm in human eeg: Frequency components, their cortical sources, and temporal behavior. *Scientific reports*, 7(1):8249.

[Bashivan et al., 2016] Bashivan, P., Rish, I., Yeasin, M., and Codella, N. (2016). Learning representations from eeg with deep recurrent-convolutional neural networks.

[Baydogan et al., 2013] Baydogan, M., Runger, G., and Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2796–2802.

[Bracewell, 2000] Bracewell, R. N. (2000). *The Fourier Transform And Its Applications*. McGraw-Hill series in electrical and computer engineering. Circuits and systems. McGraw Hill, 3rd ed edition.

[Brigham and Kumar, 2010] Brigham, K. and Kumar, B. V. K. V. (2010). Imagined Speech Classification with EEG Signals for Silent Communication: A Preliminary Investigation into Synthetic Telepathy. In *2010 4th International Conference on Bioinformatics and Biomedical Engineering*, pages 1–4. IEEE.

[Bro, 1997] Bro, R. (1997). Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171.

[Castro et al., 2018] Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. (2018). End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248.

[Chaudhry et al., 2018] Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547.

[Cheng et al., 2019] Cheng, Y., Wong, K., Hung, K., Li, W., Li, Z., and Zhang, J. (2019). Deep nearest class mean model for incremental odor classification. *IEEE Transactions on Instrumentation and Measurement*, 68(4):952–962.

[Chi et al., 2011] Chi, X., Hagedorn, J. B., Schoonover, D., and Zmura, M. D. (2011). EEG-Based Discrimination of Imagined Speech Phonemes. *International Journal of Bioelectromagnetism*, 13(4):201–206.

[Cichocki, 2013] Cichocki, A. (2013). Tensor decompositions: A new concept in brain data analysis? *Journal of SICE Control Measurement, and System Integration, special issue; Measurement of Brain Functions and Bio-Signals, 7, 507-517, (2011).*, 7.

[Cichocki et al., 2015] Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., and Phan, H. A. (2015). Tensor decompositions for signal processing applications: From two-way to multiway component analysis.

[Dalhoumi et al., 2014] Dalhoumi, S., Dray, G., and Montmain, J. (2014). Knowledge Transfer for Reducing Calibration Time in Brain-Computer Interfacing. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*.

[DaSalla et al., 2009] DaSalla, C. S., Kambara, H., Sato, M., and Koike, Y. (2009). Single-trial classification of vowel speech imagery using common spatial patterns. *Neural Networks*, 22(9):1334 – 1339. Brain-Machine Interface.

[De Vos et al., 2007] De Vos, M., De Lathauwer, L., Vanrumste, B., Van Huffel, S., and Van Paesschen, W. (2007). Canonical decomposition of ictal scalp EEG and accurate source localisation: Principles and simulation study. *Computational Intelligence and Neuroscience*, 2007.

[Deburchgraeve et al., 2009] Deburchgraeve, W., Cherian, P. J., De Vos, M., Swarte, R. M., Blok, J. H., Visser, G. H., Govaert, P., and Van Huffel, S. (2009). Neonatal seizure localization using PARAFAC decomposition. *Clinical Neurophysiology*, 120(10):1787–1796.

[Deng et al., 2013] Deng, S., Srinivasan, R., and D'Zmura, M. (2013). Cortical signatures of heard and imagined speech envelopes. Technical report, CALIFORNIA UNIV IRVINE DEPT OF COGNITIVE SCIENCES.

[Deng et al., 2010] Deng, S., Srinivasan, R., Lappas, T., and D'Zmura, M. (2010). Eeg classification of imagined syllable rhythm using hilbert spectrum methods. *Journal of neural engineering*, 7(4):046006.

[Devarajan, 2011] Devarajan, K. (2011). *Matrix and Tensor Decompositions*, pages 291–318. Springer US, Boston, MA.

[D'Zmura et al., 2009] D'Zmura, M., Deng, S., Lappas, T., Thorpe, S., and Srinivasan, R. (2009). *Toward EEG Sensing of Imagined Speech*, pages 40–48. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Farwell and Donchin, 1988] Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523.

[García-Salinas et al., 2017] García-Salinas, J. S., Villaseñor-Pineda, L., Reyes-García, C., and Torres-García, A. A. (2017). Selección de parámetros en el enfoque de bolsa de características para clasificación de habla imaginada en electroencefalogramas. *Research in Computing Science*, 140(140):123–133.

[Gepperth and Hammer, 2016] Gepperth, A. and Hammer, B. (2016). Incremental learning algorithms and applications. In *European symposium on artificial neural networks (ESANN).*

[Gu et al., 2014] Gu, S., Zhang, L., Zuo, W., and Feng, X. (2014). Projective dictionary pair learning for pattern classification. *Advances in neural information processing systems*, 27:793–801.

[Gui and Yeh, 2014] Gui, Z.-W. and Yeh, Y.-R. (2014). *Time Series Classification with Temporal Bag-of-Words Model*, pages 145–153. Springer International Publishing, Cham.

[Hao et al., 2019] Hao, Y., Fu, Y., Jiang, Y. G., and Tian, Q. (2019). An end-to-end architecture for class-incremental object detection with knowledge distillation. *Proceedings - IEEE International Conference on Multimedia and Expo*, 2019-July:1–6.

[Hasan and Roy-Chowdhury, 2014] Hasan, M. and Roy-Chowdhury, A. K. (2014). Incremental activity modeling and recognition in streaming videos. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 796–803.

[Haykin et al., 2009] Haykin, S. S. et al. (2009). *Neural networks and learning machines.* New York: Prentice Hall, 3rd edition edition.

[He and Wu, 2017] He, H. and Wu, D. (2017). Transfer learning enhanced common spatial pattern filtering for brain computer interfaces (bcis): Overview and a new approach. In *International Conference on Neural Information Processing*, pages 811–821. Springer.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

[Hitchcock, 1927] Hitchcock, F. L. (1927). The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189.

[Jayaram et al., 2016] Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., and Grosse-Wentrup, M. (2016). Transfer Learning in Brain-Computer Interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31.

[Ji et al., 2016] Ji, H., Li, J., Lu, R., Gu, R., Cao, L., and Gong, X. (2016). Eeg classification for hybrid brain-computer interface using a tensor based multiclass multimodal analysis scheme. *Computational intelligence and neuroscience*, 2016:51.

[Jia et al., 2014] Jia, X., Li, K., Li, X., and Zhang, A. (2014). A novel semi-supervised deep learning framework for affective state recognition on EEG signals. In *Proceedings - IEEE 14th International Conference on Bioinformatics and Bioengineering, BIBE 2014*, pages 30–37. IEEE.

[Kang et al., 2009] Kang, H., Nam, Y., and Choi, S. (2009). Composite common spatial pattern for subject-to-subject transfer. *IEEE Signal Processing Letters*, 16(8):683–686.

[Kaushik et al., 2019] Kaushik, P., Gupta, A., Roy, P. P., and Dogra, D. P. (2019). EEG-Based Age and Gender Prediction Using Deep BLSTM-LSTM Network Model. *IEEE Sensors Journal*, 19(7):2634–2641.

[Kiers, 1991] Kiers, H. A. L. (1991). Hierarchical relations among three-way methods. *Psychometrika*, 56(3):449–470.

[Kim et al., 2013] Kim, T., Lee, J., Choi, H., Lee, H., Kim, I. Y., and Jang, D. P. (2013). Meaning based covert speech classification for brain-computer interface based on electroencephalography. In *Neural Engineering (NER), 2013 6th International IEEE/EMBS Conference on*, pages 53–56.

[Kober et al., 2001] Kober, H., Mö, M., Nimsky, C., Rgen Vieth, J., Fahlbusch, R., and Ganslandt, O. (2001). New Approach to Localize Speech Relevant Brain Areas and Hemispheric Dominance Using Spatially Filtered Magnetoencephalography. *Hum. Brain Mapping*, 14:236–250.

[Kolda and Bader, 2009] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

[Kumar et al., 2017] Kumar, S., Sharma, A., Mamun, K., and Tsunoda, T. (2017). A Deep Learning Approach for Motor Imagery EEG Signal Classification. *Proceedings - Asia-Pacific World Congress on Computer Science and Engineering 2016 and Asia-Pacific World Congress on Engineering 2016, APWC on CSE/APWCE 2016*, pages 34–39.

[Lazebnik and Raginsky, 2009] Lazebnik, S. and Raginsky, M. (2009). Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7):1294–1309.

[Lecun et al., 2015] Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[Lee et al., 2007] Lee, H., Kim, Y.-D., Cichocki, A., and ChoI, S. (2007). Nonnegative Tensor Factorization for Continuous Eeg Classification. *International Journal of Neural Systems*, 17(04):305–317.

[Li and Zhang, 2010] Li, J. and Zhang, L. (2010). Regularized tensor discriminant analysis for single trial EEG classification in BCI. *Pattern Recognition Letters*, 31(7):619–628.

[Li and Hoiem, 2018] Li, Z. and Hoiem, D. (2018). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947.

[Liu et al., 2007] Liu, Y., Hua, S., and Weekes, B. S. (2007). Differences in neural processing between nouns and verbs in chinese: Evidence from eeg. *Brain and Language*, 103(1):75–77.

[Lotte, 2015] Lotte, F. (2015). Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain-computer interfaces. *Proceedings of the IEEE*.

[Lotte and Guan, 2010] Lotte, F. and Guan, C. (2010). Learning from other subjects helps reducing brain-computer interface calibration time. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 614–617.

[Lu et al., 2013] Lu, H., Plataniotis, K., and Venetsanopoulos, A. (2013). *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. Chapman & Hall/CRC machine learning & pattern recognition series. CRC Press.

[Lu et al., 2017] Lu, N., Li, T., Ren, X., and Miao, H. (2017). A Deep Learning Scheme for Motor Imagery Classification based on Restricted Boltzmann Machines. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(6):566–576.

[Mairal et al., 2008] Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2008). Supervised dictionary learning. *Advances in Neural Information Processing Systems*, pages 1033–1040.

[Martin et al., 2016] Martin, S., Brunner, P., Iturrate, I., Millán, J. d. R., Schalk, G., Knight, R. T., and Pasley, B. N. (2016). Word pair classification during imagined speech using direct brain recordings. *Scientific reports*, 6(1):1–12.

[Masana et al., 2020] Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and van de Weijer, J. (2020). Class-incremental learning: survey and performance evaluation. *arXiv preprint arXiv:2010.15277*.

[Matsumoto and Hori, 2014] Matsumoto, M. and Hori, J. (2014). Classification of silent speech using support vector machine and relevance vector machine. *Applied Soft Computing*, 20:95–102.

[McGuire et al., 1996] McGuire, P. K., Silbersweig, D. A., Murray, R. M., David, A. S., Frackowiak, R. S. J., and Frith, C. D. (1996). Functional anatomy of inner speech and auditory verbal imagery. *Psychological Medicine*, 26(01):29.

[Mermillod et al., 2013] Mermillod, M., Bugaiska, A., and Bonin, P. (2013). The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4:504.

[Miwakeichi et al., 2004] Miwakeichi, F., Martínez-Montes, E., Valdés-Sosa, P. A., Nishiyama, N., Mizuhara, H., and Yamaguchi, Y. (2004). Decomposing EEG data into space-time-frequency components using Parallel Factor Analysis. *NeuroImage*, 22(3):1035–1045.

[Mo et al., 2017] Mo, H., Luo, C., and Jan, G. E. (2017). EEG classification based on sparse representation. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 59–62. IEEE.

[Morioka et al., 2015] Morioka, H., Kanemura, A., ichiro Hirayama, J., Shikauchi, M., Ogawa, T., Ikeda, S., Kawanabe, M., and Ishii, S. (2015). Learning a common dictionary for subject-transfer decoding with resting calibration. *NeuroImage*, 111:167–178.

[Nguyen et al., 2017] Nguyen, C. H., Karavas, G., and Artemiadis, P. (2017). Inferring imagined speech using EEG signals: a new approach using Riemannian Manifold features. *Journal of Neural Engineering*.

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning.

[Panagopoulos, 2017] Panagopoulos, G. (2017). Multi-Task Learning for Commercial Brain Computer Interfaces. *17th International Conference on Bioinformatics and Bioengineering*, pages 86–93.

[Peng et al., 2014] Peng, Y., Meng, D., Xu, Z., Gao, C., Yang, Y., and Zhang, B. (2014). Decomposable nonlocal tensor dictionary learning for multispectral image denoising.

In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2949–2956. IEEE.

[Perani et al., 1996] Perani, D., Dehaene, S., Grassi, F., Cohen, L., Cappa, S. F., Dupoux, E., Fazio, F., and Mehler, J. (1996). Brain processing of native and foreign languages. *NeuroReport-International Journal for Rapid Communications of Research in Neuroscience*, 7(15):2439–2444.

[Plinge et al., 2014] Plinge, A., Grzeszick, R., and Fink, G. A. (2014). A bag-of-features approach to acoustic event detection. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3704–3708.

[Polikar et al., 2001] Polikar, R., Upda, L., Upda, S. S., and Honavar, V. (2001). Learn++: An incremental learning algorithm for supervised neural networks. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 31(4):497–508.

[Pressel Coretto et al., 2017] Pressel Coretto, G. A., Gareis, I. E., and Rufiner, H. L. (2017). Open access database of eeg signals recorded during imagined speech. *Proc. SPIE*, 10160.

[Quan et al., 2015] Quan, Y., Huang, Y., and Ji, H. (2015). Dynamic texture recognition via orthogonal tensor dictionary learning. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:73–81.

[Rebuffi et al., 2017] Rebuffi, S. A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). iCaRL: Incremental classifier and representation learning. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:5533–5542.

[Ren and Wu, 2014] Ren, Y. and Wu, Y. (2014). Convolutional deep belief networks for feature extraction of EEG signal. In *Proceedings of the International Joint Conference on Neural Networks*, pages 2850–2853. IEEE.

[Riaz et al., 2014] Riaz, A., Akhtar, S., Iftikhar, S., Khan, A. A., and Salman, A. (2014). Inter comparison of classification techniques for vowel speech imagery using EEG sensors. In *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, pages 712–717. IEEE.

[Roy et al., 2020] Roy, D., Panda, P., and Roy, K. (2020). Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning. *Neural Networks*, 121:148–160.

[Salama et al., 2014a] Salama, M., ElSherif, L., Lashin, H., and Gamal, T. (2014a). Recognition of unspoken words using electrode electroencephalograhic signals. In *The Sixth International Conference on Advanced Cognitive Technologies and Applications*, pages 51–5. Citeseer.

[Salama et al., 2014b] Salama, M., Lashin, H., and Gamal, T. (2014b). Recognition of unspoken words using electrode electroencephalograhic signals. *COGNITIVE 2014 : The Sixth International Conference on Advanced Cognitive Technologies and Applications*, pages 51–55.

[Sarwar et al., 2020] Sarwar, S. S., Ankit, A., and Roy, K. (2020). Incremental Learning in Deep Convolutional Neural Networks Using Partial Network Sharing. *IEEE Access*, 8:4615–4628.

[Schmidhuber, 2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117.

[Shen et al., 2017] Shen, Y., Lu, H., and Jia, J. (2017). Classification of motor imagery EEG signals with deep learning models. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10559 LNCS, pages 181–190. Springer, Cham.

[Shergill et al., 2003] Shergill, S. S., Brammer, M. J., Fukuda, R., Williams, S. C., Murray, R. M., and McGuire, P. K. (2003). Engagement of brain areas implicated in processing inner speech in people with auditory hallucinations. *British Journal of Psychiatry*, 182(JUNE):525–531.

[Suppes et al., 1997a] Suppes, P., Lin, L. Z., and Bing, H. (1997a). Brain wave recognition of words. *Proceedings of the National Academy of Sciences*, 94(26):14965 – 14969.

[Suppes et al., 1997b] Suppes, P., Lu, Z.-L., and Han, B. (1997b). Brain wave recognition of words. *Psychology*, 94:14965–14969.

[Suwicha et al., 2014] Suwicha, Pan-Ngum, S., and Israsena, P. (2014). EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation. *Scientific World Journal*, 2014:627892.

[Tabar, 2017] Tabar, Y. R. (2017). A novel deep learning approach for classification of EEG motor imagery signals. *Journal of Neural Engineering*, 14(1):016003.

[Tang et al., 2017] Tang, Z., Li, C., and Sun, S. (2017). Single-trial EEG classification of motor imagery using deep convolutional neural networks. *Optik*, 130:11–18.

[Tao et al., 2020] Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., and Gong, Y. (2020). Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192.

[Tayeb et al., 2019] Tayeb, Z., Fedjaev, J., Ghaboosi, N., Richter, C., Everding, L., Qu, X., Wu, Y., Cheng, G., and Conradt, J. (2019). Validating deep neural networks for online decoding of motor imagery movements from eeg signals. *Sensors (Switzerland)*, 19(1):1–16.

[Tibor et al., 2017] Tibor, R., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W., and Ball, T. (2017). Deep learning with convolutional neural networks for EEG decoding and visualization. *Human brain mapping*, 38(11):5391–5420.

[Torres-García et al., 2013] Torres-García, A. A., Reyes-García, C. A., L., L. V.-P., and Ramirez, J. (2013). Analisis de señales electroencefalograficas para la clasificacion de habla imaginada. *Revista mexicana de ingeniería biomedica*, 34:23 – 39.

[Torres-García et al., 2016] Torres-García, A. A., Reyes-García, C. A., Villaseñor-Pineda, L., and García-Aguilar, G. (2016). Implementing a fuzzy inference system in a multi-objective {EEG} channel selection model for imagined speech classification. *Expert Systems with Applications*, 59:1 – 12.

[Torres-García et al., 2013] Torres-García, A. A., Reyes-García, C. A., Villaseñor-Pineda, L., and Ramírez-Cortés, J. M. (2013). Análisis de señales electroencefalográficas para la clasificación de habla imaginada. *Revista Mexicana de Ingenieria Biomedica*, 34(1):23–39.

[Tosic and Frossard, 2011] Tosic, I. and Frossard, P. (2011). Dictionary Learning. *IEEE Signal Processing Magazine*, 28(2):27–38.

[Tu and Sun, 2012] Tu, W. and Sun, S. (2012). A subject transfer framework for EEG classification. *Neurocomputing*, 82:109–116.

[Wang et al., 2013a] Wang, J., Liu, P., She, M. F., Nahavandi, S., and Kouzani, A. (2013a). Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control*, 8(6):634 – 644.

[Wang et al., 2013b] Wang, J., She, M., Nahavandi, S., and Kouzani, A. (2013b). Human identification from ecg signals via sparse representation of local segments. *IEEE Signal Processing Letters*, 20(10):937–940.

[Wang et al., 2017] Wang, K., Wang, X., and Li, G. (2017). Simulation experiment of bci based on imagined speech eeg decoding. *arXiv preprint arXiv:1705.07771*.

[Wang et al., 2013c] Wang, L., Zhang, X., Zhong, X., and Zhang, Y. (2013c). Analysis and classification of speech imagery EEG for BCI. *Biomedical Signal Processing and Control*, 8:901–908.

[Waytowich et al., 2016] Waytowich, N. R., Lawhern, V. J., Bohannon, A. W., Ball, K. R., and Lance, B. J. (2016). Spectral transfer learning using information geometry for a user-independent brain-computer interface. *Frontiers in Neuroscience*, 10(SEP).

[Wei et al., 2018] Wei, C.-S., Lin, Y.-P., Wang, Y.-T., Lin, C.-T., and Jung, T.-P. (2018). A subject-transfer framework for obviating inter- and intra-subject variability in eeg-based drowsiness detection. *NeuroImage*, 174:407 – 419.

[Weis et al., 2009] Weis, M., Romer, F., Haardt, M., Jannek, D., and Husar, P. (2009). Multi-dimensional space-time-frequency component analysis of event related EEG data using closed-form PARAFAC. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 349–352.

[Wronkiewicz et al., 2015] Wronkiewicz, M., Larson, E., and Lee, A. K. C. (2015). Leveraging anatomical information to improve transfer learning in brain computer interfaces. *Journal of Neural Engineering*, 12(4):046027.

[Wu et al., 2014] Wu, D., Lance, B., and Lawhern, V. (2014). Transfer learning and active transfer learning for reducing calibration data in single-trial classification of visually-evoked potentials. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2801–2807. IEEE.

[Wymbs et al., 2013] Wymbs, N. F., Ingham, R. J., Ingham, J. C., Paolini, K. E., and Grafton, S. T. (2013). Individual differences in neural regions functionally related to real and imagined stuttering. *Brain and Language*, 124(2):153–164.

[Xia et al., 2016] Xia, Q., Wang, L., and Peng, G. (2016). Nouns and verbs in chinese are processed differently: Evidence from an erp study on monosyllabic and disyllabic word processing. *Journal of Neurolinguistics*, 40:66–78.

[Xiao et al., 2014] Xiao, T., Zhang, J., Yang, K., Peng, Y., and Zhang, Z. (2014). Error-driven incremental learning in deep convolutional neural network for large-scale image classification. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 177–186, New York, NY, USA. ACM.

[Yang et al., 2015] Yang, H., Sakhavi, S., Ang, K. K., and Guan, C. (2015). On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, volume 2015-Novem, pages 2620–2623. IEEE.

[Ye and Zhu, 2019] Ye, X. and Zhu, Q. (2019). Class-Incremental Learning Based on Feature Extraction of CNN with Optimized Softmax and One-Class Classifiers. *IEEE Access*, 7(c):42024–42031.

[Zhang et al., 2017] Zhang, J., Yan, C., and Gong, X. (2017). Deep convolutional neural network for decoding motor imagery based brain computer interface. *2017 IEEE International Conference on Signal Processing, Communications and Computing, ICSPCC 2017*, 2017-Janua:1–5.

[Zhang et al., 2020] Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., and Kuo, C. C. J. (2020). Class-incremental learning via deep model consolidation.

[Zhang et al., 2019] Zhang, Y., Zhang, X., Sun, H., Fan, Z., and Zhong, X. (2019). Portable brain-computer interface based on novel convolutional neural network. *Computers in Biology and Medicine*, 107:248–256.

[Zhao et al., 2009] Zhao, Q., Caiafa, C. F., Cichocki, A., Zhang, L., and Phan, A. H. (2009). Slice oriented tensor decomposition of EEG data for feature extraction in space, frequency and time domains. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5863 LNCS(PART 1):221–228.

[Zhao and Rudzicz, 2015] Zhao, S. and Rudzicz, F. (2015). Classifying phonological categories in imagined and articulated speech. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 992–996.

[Zhou et al., 2012] Zhou, W., Yang, Y., and Yu, Z. (2012). Discriminative dictionary learning for EEG signal classification in Brain-computer interface. In *2012 12th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1582–1585. IEEE.

[Zhuolin Jiang et al., 2013] Zhuolin Jiang, Zhe Lin, and Davis, L. S. (2013). Label Consistent K-SVD: Learning a Discriminative Dictionary for Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2651–2664.

[Zubair and Wang, 2013] Zubair, S. and Wang, W. (2013). Tensor dictionary learning with sparse tucker decomposition. In *2013 18th International Conference on Digital Signal Processing, DSP 2013*, pages 1–6. IEEE.