Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

InIT Institut für angewandte
Informationstechnologie

Projektarbeit (Informatik)

From human to artificial neural networks

Autoren	Maximilian Bürgi Newroz Gilgil
Hauptbetreuung	Ricardo Chavarriaga Thilo Stadelmann
Datum	19.12.2020

Abstract

With increasing interest in BCI (brain-computer interface) more and more research is done on EEG (electroencephalogram) decoding. Particularly in recent years there is an ever-growing endeavour to utilize ConvNets (convolutional neural networks) to decode EEG data because of its utility to self extract features and its relative efficient training compared to simple ANNs (artificial neural networks). But many questions arise. What kind of ConvNet architectures are good? How high are the generalization capacities of such ConvNets and how interpretable are these processes? Following these questions, this thesis makes a comparison between three recently proposed ConvNet architectures. This thesis compares their performance and also analyzes the drop in performance if they are confronted with EEG data from an unknown subject and it takes a short look into the interpretability of these systems with a visualization approach of their respective feature selection.

Experiments are conducted over different publicly available MI (motor imagery) datasets, which show that there is a substantial performance drop in accuracy for EEG data of previously unseen subjects. Methods were applied to visualize that these three models extract similar features, which are interpretable by humans.

Preface

Intelligence is one of the many wonders of nature, it is anchored in our mind and it guides us to make the right decisions. It crystallizes in the brains of living beings, and to this day it is not possible for us to get an exact insight into the functioning of this black box. As students of ZHAW we are thrilled to have the opportunity to work on such a state-of-the-art topic. This project was a great chance for us to educate ourselves in the topics of Neuroinformatics and Deep Learning and to get hands-on experience.

We would like to extend special thanks to our supervisors Prof. Dr. Thilo Stadelmann and Dr. Ricardo Chavarriaga. Their cooperation and support were a joy. Further thanks goes to the research team of the Neuromedical AI Lab at the Albert-Ludwigs-University Freiburg for providing us with the tools to apply deep learning to EEG decoding with "Braindecode".

Zürcher Hochschule
für Angewandte Wissenschaften

zh
aw

School of
Engineering

Erklärung betreffend das selbständige Verfassen einer Projektarbeit an der School of Engineering

Mit der Abgabe dieser Projektarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Projektarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

Zürich, 18.12.2020

Unterschriften:

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Projektarbeiten zu Beginn der Dokumentation nach dem Abstract bzw. dem Management Summary mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

Contents

1. Introduction 1
1.1. Motivation 1
1.2. Project objective 2
1.3. Overview 2

2. Foundations & Related Work 3
2.1. ANN: Artificial neural network 3
2.1.1. ConvNet: Convolutional Neural network 4
2.2. EEG: Electroencephalogram 5
2.2.1. MI: Motor Imagery 6
2.3. EEG Decoding Methods 6
2.3.1. FBCSP - Filter Bank Common Spatial Pattern 7
2.3.2. ConvNets applied to EEG 7

3. Methods 8
3.1. **ConvNet Design Choices** 8
3.2. Braindecode - Toolbox for EEG Decoding 9
3.2.1. EEGNetv4 9
3.2.2. ShallowFBCSPNet 10
3.2.3. Deep4Net 11
3.3. Datasets 12
3.3.1. BNCI2014001 12
3.3.2. BNCI2014002 12
3.3.3. BCI2014004 12
3.3.4. BNCI2015001 12
3.4. Pipeline 13
3.5. Metrics 14
3.6. Visualization: Input-perturbation network-prediction correlation map 15

4. Experiments 16
4.1. Model Comparison 16
4.1.1. Deep4Net is the best model for inter- and intra-subject classification 16
4.1.2. The ConvNets on average have a 38.7% performance drop for inter-subject classification compared to intra-subject classification 18
4.1.3. The ConvNet models produce similar correlation maps 18
4.2. Subject Comparison 19
4.2.1. Subjects with underwhelming classification accuracy have atypical correlation maps 19
4.2.2. Confusion matrices of badly performing subjects overfocus on a subset of classes 20

5. Discussion and Outlook 21
5.1. Critique 21
5.2. Model Comparison 21
5.3. Subjects 21
5.4. Conclusion 22
5.5. Follow Up 22

6. Verzeichnisse 23
Literaturverzeichnis 24
Abbildungsverzeichnis 27

Tabellenverzeichnis 28

A. Appendix **I**

A.1. Environment Setup I

A.1.1. GPU cluster I

A.1.2. Software I

A.1.3. Running with Docker II

A.1.4. Dockerfile II

A.2. Model Comparison - Other Metrics III

1. Introduction

1.1. Motivation

Humanity has been fascinated for ages with seemingly outlandish predictions about what science (and/or magic) would enable us to do in the future and imagining a utopia around those claims. Since the 20th century this is called science fiction but the idea behind it has been around for way longer. One of the earliest known examples of such a future vision is the short novel "True History" by the 2nd-century-born Assyrian satirist Lucian of Samosata, where he describes a voyage to the moon long before even human flight was invented[1]. While this example also incorporates many scientific inaccuracies it has to be considered that it is hard to predict what will be actually possible in the future and what might actually some day become a reality. A more modern example would be "Frankenstein, or, The Modern Prometheus" from 1818 by Mary Shelley[2] where the protagonist, a type of scientist (before the word was in use), uses a disembodied brain to control a complete human body. This would be the earliest conception of the idea of what we nowadays might call a brain computer interface (BCI), the idea of "wiring up" a brain with a whole body which provides an interface for the brain to interact with the world around it (even though technically a body wouldn't be called a computer). Modern science fiction goes even further with ideas of telepathy or machine-enhanced humans.

Though none of these ideas are actually reality today (except that we landed on the moon) scientists sure are touching on the topic of how to interface between a brain and a computer. With the discovery of electroencephalography (EEG) in the 20th century, a way to read "brain waves" from the surface of a human head, the modern field of brain-computer interface (BCI) was born in the late half of the century. This field concerning direct communication between a brain and an external device might actually make these visions from science fiction reality one day. Surprisingly, one of the first publicly famous applications of BCI wasn't performed by a scientist but by a musician in 1965. A composer of experimental music, Alvin Lucier, created a musical piece called "Music for Solo Performer" where he amplified alpha brain waves (brain waves of a certain frequency) to excite percussion instruments to generate sound while sitting still in a chair[3]. While not being scientifically relevant and not accurately interpreting the information encoded in brain waves, his performance illustrated the possibility to read brain activity and use it to influence the outside world.

Today BCI is an ever growing topic in research and its use in many different applications is being considered. These applications involve generating assistive devices (communication or mobility aids for people with disabilities[4]) or even human enhancement (neuroergonomics, neuromarketing, educational and security-related applications[5]). Since 2016 there is even an international competition called "Cybathlon" organized by ETHZ in which people with physical disabilities compete in multiple disciplines against each other using assistive devices (many incorporating some kind of BCI) and showcasing new research on assistance systems.

It is one step to measure electrical signals from the brain but the next one would be to process and interpret these signals which is what this project will concern itself with. As opposed to traditional methods to decode EEG data which work in several steps (preprocessing, feature extraction, classification) and often require qualified personnel to handcraft features to focus on, there is a novel approach since a few years to apply convolutional neural networks (ConvNets) from the field of deep learning (DL) to EEG data analysis[6]. Deep learning concerns itself with the modelling of neural networks inspired by the human brain to facilitate learning and convolutional neural networks are promising DL algorithms which have been very successful since their conception to process sensory data especially in image recognition. Applying ConvNets to EEG decoding provides an all-in-one method that incorporates all steps of traditional EEG decoding methods and is usable without expert knowledge about EEG data. The biggest problem is the lack of sufficient training data to train such sophisticated models. This approach currently seems very promising and could have many groundbreaking applications in the near future. Disorders affecting the brain could be detected more efficiently and at lower cost. People who have lost joints could move their prostheses with unprecedented efficiency. It would even be possible to communicate telepathically with machines.

This project aims to further the knowledge in interpreting EEG data with ConvNets, provide an overview over different approaches on how to decode that data and to overall help to bridge the gap between a human neural network and an artificial one.

1.2. Project objective

A lot of questions about EEG decoding with ConvNets still remain open to this day.

Intra-subject classification, which means that the ConvNet is evaluated on the same subjects as it was trained on (but not the same data), is a well explored topic while inter-subject classification, which is classification that is evaluated on new subjects that weren't trained on, is less explored. This is because inter-subject classification is the harder problem because there are bigger differences between EEG patterns across different subjects as opposed to between different sessions with the same subject. This project will take a look at the drop in performance from intra- to inter-subject classification and whether anything can be done to minimize that drop.

Another interesting question is the one about what a ConvNet does with the data that it is provided with. A general ConvNet is considered as a "black box" that produces magic results with astonishing performance and it is hard to look inside a ConvNet and make sense of all the numbers clustered together. A second goal of this project will be to figure out how to visualize what a ConvNet has learned after training.

Lastly a finished ConvNet should be able to be used on many different subjects outside of laboratory conditions and there it does not have any data on newly trained subjects. This project will examine how long a well-trained model needs to be trained on a new subject to perform reasonably well.

There are many different BCI paradigms which are different types of tasks that are issued to subjects during an experiment. This project will consider motor imagery (MI) as it is one of the more widely used BCI paradigms. Motor imagery is the process of imagining the movement of a limb without actually moving it.

The main objectives of this project are therefore the following questions:

- How much performance is lost when applying a well-performing intra-subject ConvNet architecture to produce inter-subject predictions?
- Can the features extracted by a ConvNet be interpreted in a reasonable manner by visualization?
- How many more trials of a new subject does a pretrained model require to reach similar performance as for subjects that it is trained on?

1.3. Overview

This report is structured in four main parts. Initially in chapter 2 a theoretical basis is established to provide the reader with the required knowledge to understand the project. In the second part in chapter 3 used tools and implementations are described to reproduce the project. The third part in chapter 4 describes the conducted experiments and results. And lastly in chapter 5 the results are discussed. In the future the gained knowledge of this project can provide a basis to conduct a more efficient and precise analysis of brain activity with a ConvNet approach.

2. Foundations & Related Work

First in section 2.1 a brief overview of artificial neural networks (ANN) is given. Afterwards in subsection 2.1.1 the fundamental concepts of convolutional neural networks (ConvNets) are explained. In section 2.2 the basic concepts of electroencephalogram (EEG) are provided, these are then deepened in the BCI paradigms motor imagery in subsection 2.2.1. In section 2.3 EEG decoding methods are explained and the most relevant ones are elaborated.

2.1. ANN: Artificial neural network

Artificial neural networks are algorithmic models of interconnected neurons. This concept is vaguely based on the biological neural networks found in human brains. ANN are made up of different layers of artificial neurons, also called units. These neurons are interconnected with all neurons of the prior and posterior layer. These connections are called synapses. The first layer is called “input layer”, the last is called “output layer”, layers in between are called “hidden layers”.

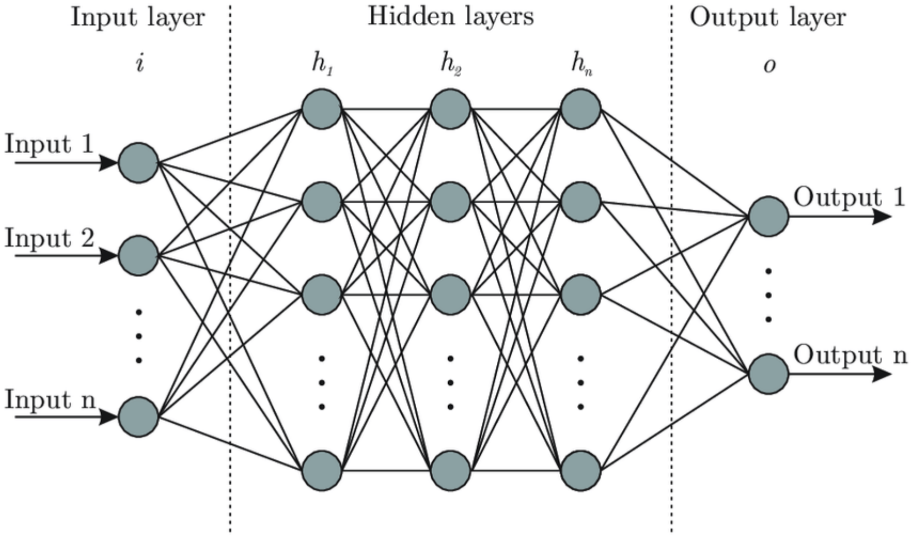

Figure 2.1.: Artificial Neural Network[7]

Each synapse between neurons is weighted. So each neuron gets a weighted sum of the signals from the prior layer. Each neuron has its unique bias, which will get added to the sum of the weighted inputs. This sum will be passed to an activation function. We can think of the activation function as a mathematical function that normalises the output of that neuron. The result of the activation function will then be passed to the neurons on the next layer.

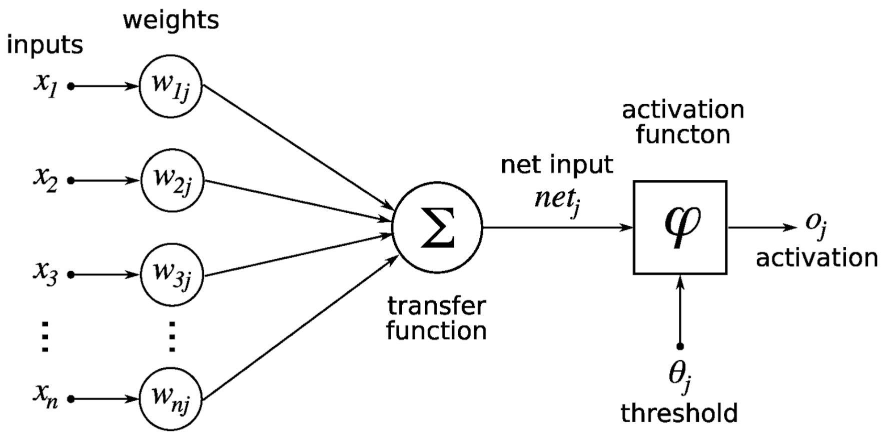

Figure 2.2.: Neuron inside a ANN[8]

2.1.1. ConvNet: Convolutional Neural network

A ConvNet, also called CNN, is a convolving ANN that is primarily used for image recognition tasks. Each input in a ConvNet will pass through a series of convolutional layers and pooling layers which transform the input until it gets to the end of feature extraction, where it gets passed as a vector to the fully connected layer which classifies the data. ConvNets are a major improvement to traditional neural networks in regards to processing data where spatial and temporal (space and time dimensional) information will get lost if the input data is flattened to a 1-dimensional vector.

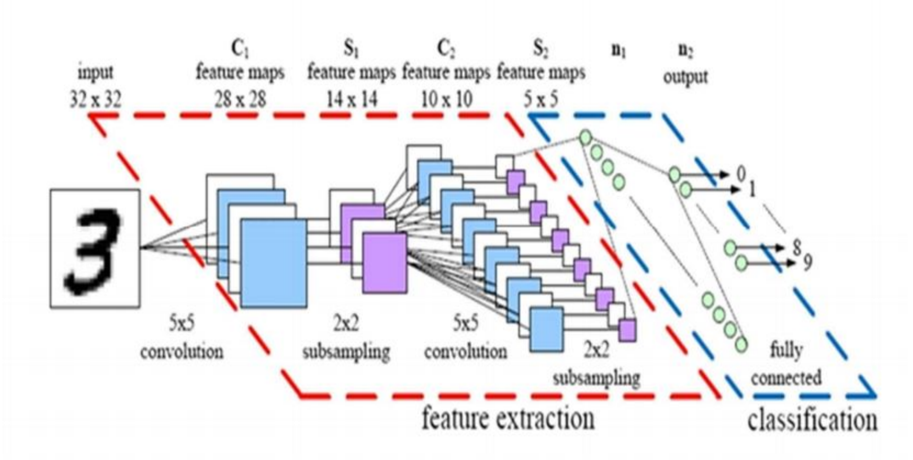

Figure 2.3.: Convolutional Neural Network[9]

Convolutional layer

The convolutional layer convolves a 2D or 3D matrix which it gets as input. This means that a filter (kernel) is applied to the input. The kernel can be seen as the weights of that layer. The kernel is a matrix which gets iterated over the input. While iterating the component wise products of the matched part of the matrix and the filter are calculated, the summation of these products is the value of the new field. As shown in the following image.

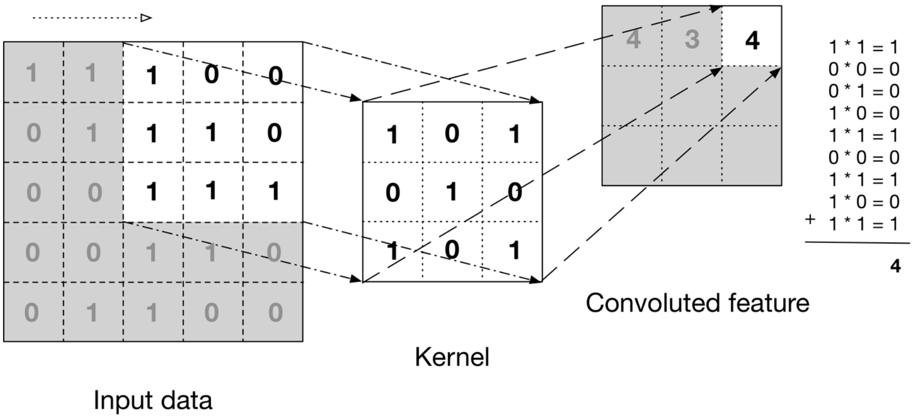

Figure 2.4.: Convolution Illustration[10]

A convolutional layer usually contains several different filters that are each applied to each input thus multiplying the amount of data. With m inputs and n kernels the convolutional layer returns $m * n$ outputs (called feature maps) that are each a little smaller than the input (by the kernel size in the corresponding dimension). The shrinking of each input is due to how a convolution operates.

Pooling layer

Pooling layers downsample the data from the previous layer by discarding most values except the maximal one in each patch of the input where the patch size is predefined (e.g. 2x2 in Figure 2.5).

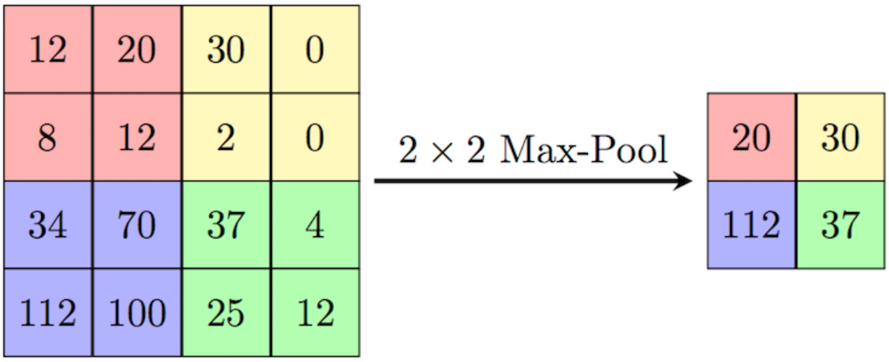

Figure 2.5.: Max-Pooling Illustration[11]

Fully Connected Layer

The pooling and convolutions on the prior layers reduce the input to a vector. This vector is passed to the connected layer where it finally gets classified. The connection layers are a simple ANN.

2.2. EEG: Electroencephalogram

EEG is an imaging method for the depiction of electrical activity on the scalp generated by the brain. For this purpose, various electrodes (sensors) are mounted on the surface of the head, directly on the cortical surface (electrocorticogram) or under the cortical surface (electrogram) of the subject. For this project we will examine EEG measured from the scalp surface. In this way EEG is an unobtrusive and non-invasive procedure.

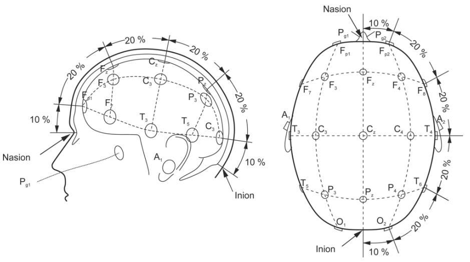

Figure 2.6.: EEG Sensors on Subject[12]

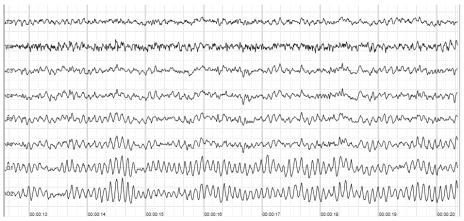

Figure 2.7.: Visualization of EEG Data. Each line represents the signal of a sensor during a whole trial.[13]

Brainwaves are categorized into four basic groups[14]:

- Delta 0.1-4 Hz,
- Theta 4-8 Hz,
- Alpha 8-13 Hz,
- Beta 13-30 Hz,
- Gamma 30-100 Hz

This procedure has its limitations. Since the electrodes are placed onto the surface of the head, the electrical activity recorded is mostly a summation of excitatory and inhibitory postsynaptic potentials on the superficial layers of the cortex. Propagation of electrical activity can give a misleading impression of the exhibiting location[15].

2.2.1. MI: Motor Imagery

Motor imagery (MI) describes the task of a subject to imagine the movement of a limb. During motor imagery oscillatory events are picked up by EEG sensors from brain areas associated with preparation, control and carrying out of voluntary motion. Typical movement classes are movements by left hand, right hand, feet and of the tongue "since these events have been shown to produce significant and discriminative changes in the EEG signals relative to background EEG"[16]. The most relevant frequency bands for motor imagery are alpha, beta and gamma, whereas gamma is sometimes omitted since it is likely affected by undesired artifacts and hard to detect from the scalp surface[17]. Because of this most MI-related BCI algorithms focus on the frequency bands from 8 to 30 Hz (alpha beta frequency band).

A typical experiment setup to collect MI data is performed in a non-distracting environment where a subject is sitting on a comfortable chair in front of a computer screen[18]. The experiment is split into trials. A trial is a time frame in which EEG data is recorded which should correlate with MI. At the start of such a trial, a fixation cross appears on the aforementioned screen to grab the subjects attention, often followed by an acoustic sound. After a time frame the required MI action is shown on the screen, to prompt the subject to perform the desired motor imagery task[19;20;21;22]. Often the subjects are closely observed to prevent body movements, if such a noise is detected a trial is either dropped or repeated. Depending on the experiment the subjects will be shown an indication of their performance.

2.3. EEG Decoding Methods

Most EEG decoding algorithms usually include two steps, feature extraction followed by feature classification. Before classification the feature extraction is applied because of the low signal-to-noise ratio of EEG data. Otherwise a classification leads to a low accuracy by overfitting to noise or data from irrelevant channels. There are many different feature extraction algorithms in use as for example wavelet transform[23], common spatial patterns (CSP)[24], empirical mode decomposition[25], and so on. Filter-bank CSP (FBCSP), a variation of CSP, is currently one of the best-performing EEG decoding methods[26]. As opposed to those traditional EEG decoding algorithms researchers have been experimenting with a new approach by applying deep learning techniques that are used successfully in image recognition on complex brain function signals. Convolutional neural networks (ConvNets) are the most widely used out of these techniques because of their outstanding performance in feature learning from images[29]. Their main advantage compared to traditional methods is that it implements feature learning and classification simultaneously and can extract features from across all input data as opposed to FBCSP which can only inspect predefined frequency bands separated from each other[30].

2.3.1. FBCSP - Filter Bank Common Spatial Pattern

The FBCSP algorithm is an established method to classify motor imaginary EEG data[26]. Its effectiveness is dependent on the subject-specific frequency band.

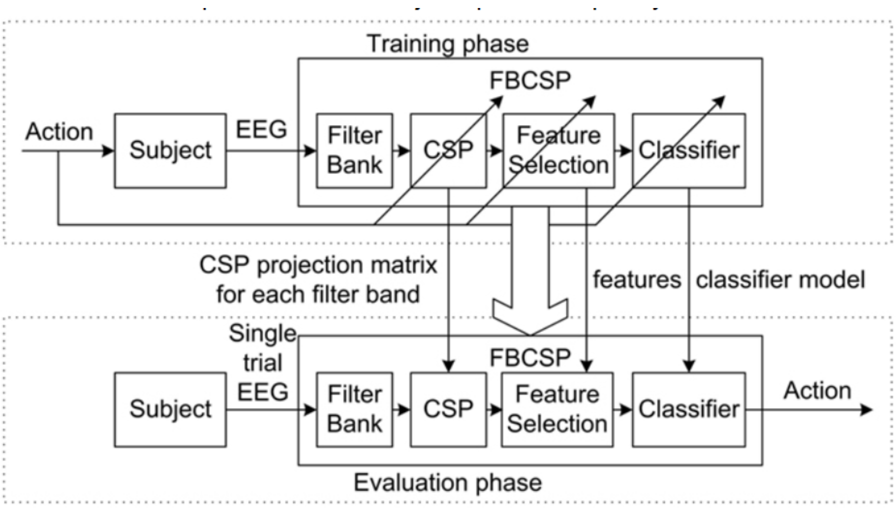

Figure 2.8.: Architecture of the filter bank common spatial pattern (FBCSP) algorithm for the training and evaluation phases.[27]

The FBCSP has 4 stages of signal processing and ML[28]:

1. Band-Pass Filtering: Decomposition of EEG data into frequency pass bands.
2. Spatial Filtering: Spatial filtering using the CSP algorithm to extract features based on Desynchronization/Synchronization.
3. Feature Selection: Selecting appropriate features for classification, various feature selection algorithms are applicable.
4. Classification: A classifier is trained to classify the trial: In the case of Braindecode regularized linear discriminant analysis (RLDA) classifier.

2.3.2. ConvNets applied to EEG

Compared to a ConvNet in computer vision (see section 2.1.1), an important design choice for a ConvNet in EEG Decoding is how to represent the input data to the network[31]. Different approaches are used for this task such as converting an EEG temporal series into 2D images by applying the short time Fourier transform method by Tabar and Halici[32] or transforming EEG data in a series of topology-preserving multi-spectral images to preserve spatial information that is lost on standard EEG analysis by Bashivan et al[33].

3. Methods

This chapter is about the applied tools and algorithms to conduct the experiments. In section 3.1 design choices about the used ConvNet models are described. Section 3.2 explains the use of "Braindecode" and the ConvNet models that are used in this project are discussed. Section 3.4 elaborates the pipeline that was used for the experiments. The used metrics to measure the experiments performance are described in section 3.5. And finally section 3.6 goes into detail about how to visualize a trained model. In the Appendix in section A.1 a description of the project setup can also be found.

3.1. ConvNet Design Choices

Input representation

As described in section 2.3.2 there are different approaches on how to handle the EEG input to a ConvNet. In this project ConvNets are considered which "represent the input as a 2D-array with the number of time steps as the width and the number of electrodes as the height"[31]. This approach has been shown to at least match and slightly outperform the standard baseline approach with FBSCP and also to significantly reduce the computational complexity by reducing the input dimensionality compared with the "EEG-as-an-image" approach[31].

Sampling strategy

ConvNets have an inherit disadvantage in that they require a large amount of training data to achieve usable results. Data for EEG decoding is usually rather small in comparison to image recognition tasks where a model can be trained on millions of images. Therefore, training is not just done on whole trials but each crop is cut into multiple crops to multiply the input data. With regular trialwise decoding (see Fig. 3.1) a complete trial is pushed through the network producing one single prediction with which the loss is calculated. With cropped decoding (see Fig. 3.2) a trial is cut into crops that are each pushed through the network. Each crop produces a prediction and the average of all predictions is used to compute the loss. To reduce the computational burden from cropped decoding multiple neighbouring crops (called a compute window) can be pushed through the network simultaneously. This has been shown to improve accuracy on frequencies higher than 4 Hz and will therefore be used in this project.[31]

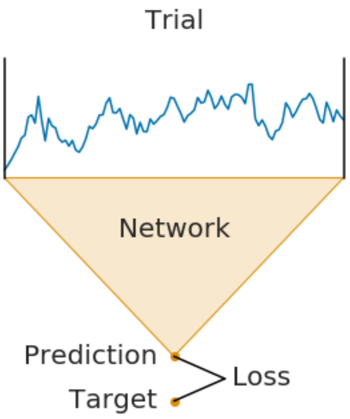

Figure 3.1.: Trialwise Decoding[31]

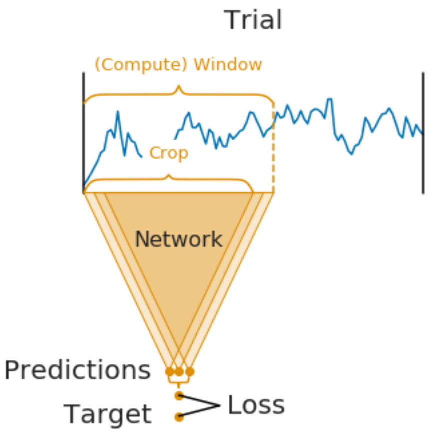

Figure 3.2.: Cropped Decoding[31]

3.2. Braindecode - Toolbox for EEG Decoding

This project is built on the code developed by the Neuromedical AI Lab at the Albert-Ludwigs-University Freiburg called “Braindecode”. This toolbox offers a simple way to construct pipelines from loading and preprocessing a dataset all the way to evaluating a trained model (see steps in section 3.4 "Pipeline"). It therefore provides a straightforward way to conduct experiments on EEG data. It offers a lot of utilities to apply deep learning methods on EEG decoding with a focus on ConvNets. The code was developed in conjunction with the paper “Deep learning with convolutional neural networks for EEG decoding and visualization” which aimed to compare the performance of ConvNets to the widely used filter bank common spatial patterns (FBCSP) algorithm for end-to-end EEG decoding.

At the time of this project "Braindecode" offers seven ConvNet models where three of them contain bugs that require advanced machine learning knowledge to resolve and one is of an old version and therefore these weren’t used in this project. The remaining three models were all used to provide a good overview over different types of ConvNets from shallow to deep and from generic to specific in respect to EEG decoding. The following models were used:

- EEGNetv4 (3.2.1 A deep ConvNet tailored to be applicable to EEG data concerning many different BCI paradigms and to produce neurophysiologically interpretable features.
- ShallowFBCSPNet (3.2.2) A shallow ConvNet implementation inspired by the FBCSP pipeline, a well-performing algorithm specifically tailored to EEG data.
- Deep4Net (3.2.3 A deep ConvNet inspired by successful architectures from computer vision with a generic approach to feature extraction.

The following subsections explain the used models in detail.

3.2.1. EEGNetv4

EEGNetv4 is designed to use as few parameters as possible in the model which makes it a good candidate to compare to Deep4Net as a computationally more efficient version of a deep ConvNet. The detailed implementation is given by V. Lawhern et al.[34] and illustrated in figure 3.3.

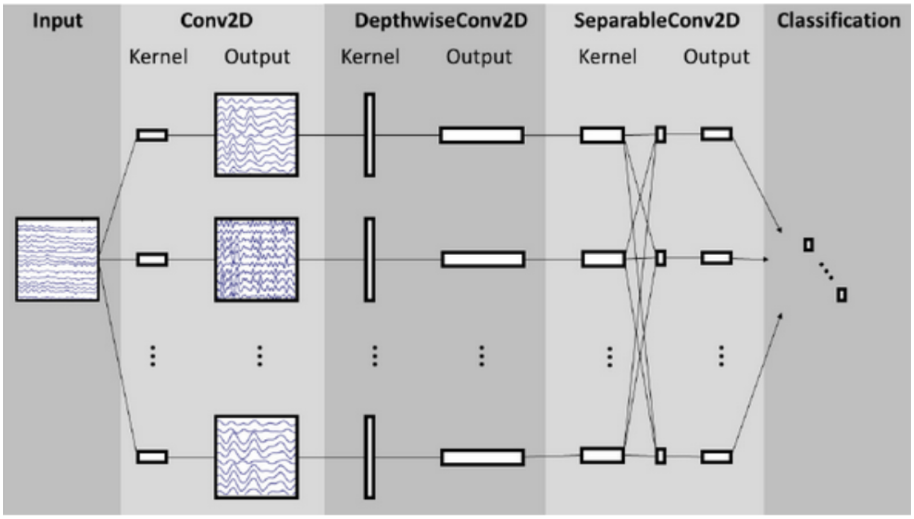

Figure 3.3.: "Overall visualization of the EEGNet architecture. Lines denote the convolutional kernel connectivity between inputs and outputs (called feature maps) . The network starts with a temporal convolution (second column) to learn frequency filters, then uses a depthwise convolution (middle column), connected to each feature map individually, to learn frequency-specific spatial filters. The separable convolution (fourth column) is a combination of a depthwise convolution, which learns a temporal summary for each feature map individually, followed by a pointwise convolution, which learns how to optimally mix the feature maps together.”[34]

3.2.2. ShallowFBCSPNet

The layers of this network can be compared to the steps of FBCSP (see 2.3.1). They are illustrated in figure 3.4. The first layer performs a temporal convolution analogous to the bandpass filter step in FBCSP and the second layer mimics FBCSP's CSP spatial filter. After that a squaring nonlinearity, a mean pooling layer and a logarithmic activation function are applied corresponding to the feature selection with trial log-variance computation in FBCSP. The shallow FBCSP has a big advantage over regular FBCSP by consisting of only a single network and not a collection of separated steps and by that it can optimize all "FBCSP steps" jointly. A more detailed description is given by Schirrmeister et al.[35].

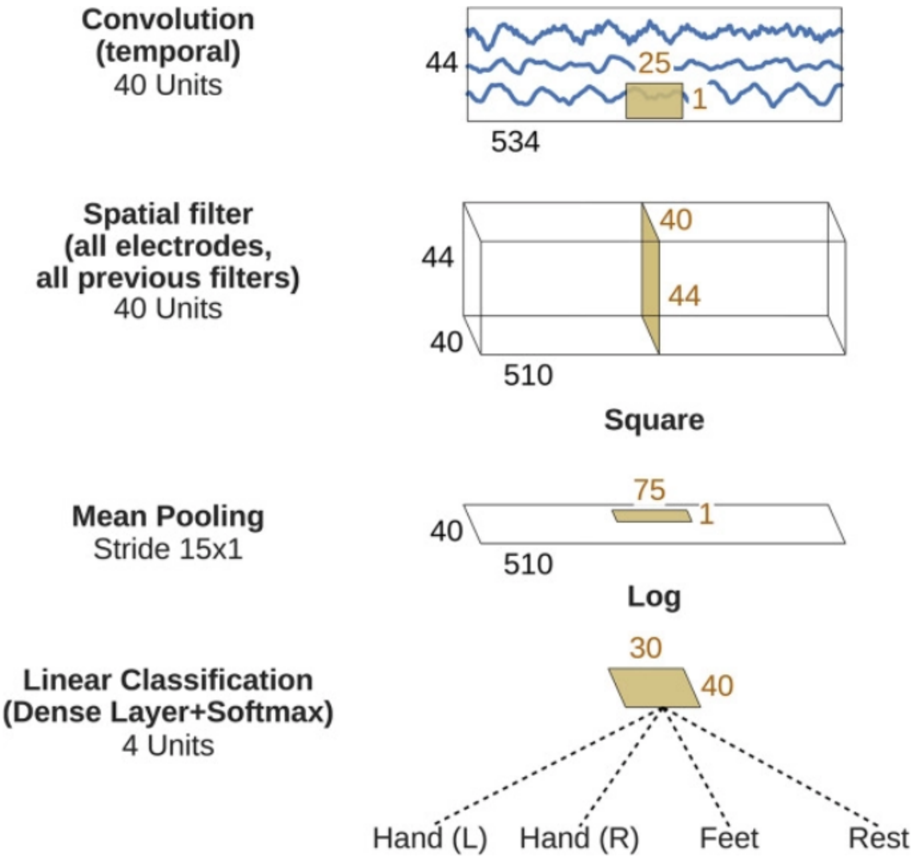

Figure 3.4.: Architecture of the shallow ConvNet designed by Schirrmeister et al[31].
EEG input (at the top) is progressively transformed toward the bottom, until the final classifier output. Black cuboids: inputs/feature maps; brown cuboids: convolution/pooling kernels. The corresponding sizes are indicated in black and brown, respectively. Sizes are for the cropped training version, see the section "Architecture differences." Each spatial filter has weights for all possible pairs of electrodes with filters of the preceding temporal convolution. Note that in these schematics, proportions of maps and kernels are only approximate.

This project will consider this model as a baseline for performance reached with non-ConvNet approaches as it performs at least as good as regular FBCSP[31].

3.2.3. Deep4Net

Deep4Net uses a generic feature extraction approach to avoid being restricted to specific feature types. This is done to improve performance and automate the feature extraction which enables the network to do without predefined feature types determined by an expert as in FBCSP which uses spectral power modulations. Furthermore a generic approach can easily be extended to utilize future developments in deep learning. The network consists of four convolution-max-pooling blocks and a dense softmax classification layer as illustrated in figure 3.5. The first block is different from the other three by applying a temporal and a spatial filter to handle EEG input. A more detailed description in given by Schirrmeister et al.[31].

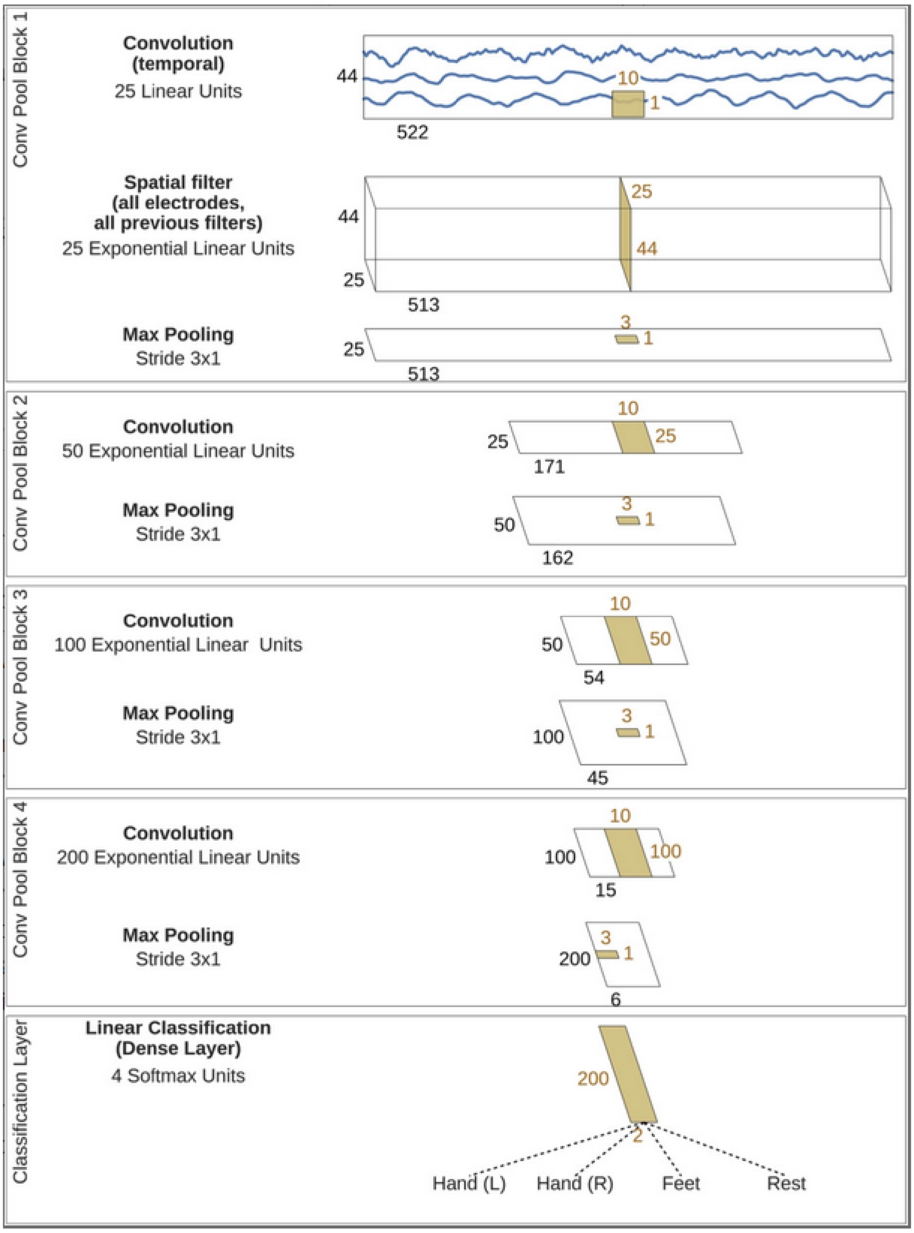

Figure 3.5.: Architecture of the Deep4 ConvNet designed by Schirrmeister et al[31]. Same conventions as in figure 3.4

3.3. Datasets

For our experiments we use the BCI Competition datasets, because they are publicly available and widely used, this enables high comparability and reproducibility. Each dataset has two or four classes and between 3 to 22 input channels for their respective measurement. The data sets consist of single-trials of spontaneous brain activity. We used the following data sets:

3.3.1. BNCI2014001[19]

Subjects: 9
Classes: 4, left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4)
Channels: 22
Structure: For each subject two sessions on different days were recorded. Each session consists of 6 runs. Each run is separated by short breaks and consist of 48 trials, 12 for each class, making a total of 288 trials.
Procedure: t = 0s: fixation cross on black screen, short acoustic warning tone
t = 2s: for 1.25 seconds the class was shown
t = (2,6): subject does motor imaginary
t = 6s: disappearance of the fixation cross

3.3.2. BNCI2014002[20]

Subjects: 13
Classes: 2, right hand (class 1), both feet (class 2),
Channels: 15
Structure: For each subject there was one session. The session is comprised of eight runs. Each run is separated by short break of random length between two to three seconds and consist of 20 trias, making 80 for each class, leading to a total of 160 trials.
Procedure: t = 0s: fixation cross on black screen, short acoustic warning tone
t = 3s: for 1 seconds the class is shown
t = (4,8): subject does motor imaginary
t = 6s: disappearance of the fixation cross

3.3.3. BCI2014004[21]

Subjects: 9
Classes: 2, left hand (class 1), right hand (class 2)
Channels: 3
Structure: For each subject there were five sessions recorded. Every session is comprised of six runs. Each run is separated by short break of random length between 1.5 to two second and consists of ten trials per class, making 60 for each class per session, leading to a total of 6'480 trials.
Procedure: t = 0s: fixation cross on black screen, short acoustic warning tone
t = x (random): for 1.25 seconds the class is shown
t = (x+1.25,x+5.25): subject does motor imaginary
t = 6s: disappearance of the fixation cross

3.3.4. BNCI2015001[22]

Subjects: 12
Classes: 2, right hand (class 1), feet(class 2)
Channels: 3
Structure: For each subject there were 3 sessions, each session consisting of 5 runs, were as one run has 20 trials for each class, making a total of 7'800 trials.
Procedure: t = 0s:start reference period
t = 3s: for 1.25 seconds the class is shown
t = 4.25: subject does motor imaginary
t = 8s: end of trial

3.4. Pipeline

With Braindecode[31] a pipeline was developed that covers all steps of EEG decoding from the data loading until the validation. The pipeline consists of the following steps:

- **Preprocessing**
  The desired dataset is loaded from the MOABB database and preprocessing is applied. All other sensor data (e.g. Magnetoencephalography - MEG) is discarded except EEG, the data is bandpass filtered for globally defined frequencies and exponential moving standardization is applied.
  The bandpass filtering is needed to compare specific frequency bands and to filter out very high frequencies to reduce noise. For example when comparing a ConvNet to FBCSP the dataset is usually filtered to below 38 Hz such that both approaches have access to the same data and no frequencies are filtered out by FBCSP that the ConvNet can use[35].
  The exponential moving standardization is applied as a normalization of the input data. This is done to combat the phenomenon called *internal covariate shift* which is "the change in the distribution of network activations due to the change in network parameters during training" (Ioffe et al[36]) which would lead to significantly longer training durations.
- **Create model and compute windowing parameters**
  The desired model is created and each trial is cut into windows which will be used for cropped decoding (see section 3.1) in the training phase. This is done over all experiments as a more data-efficient approach to tackle the problem of the small amount of input data. A predefined random seed is set for the numpy random module and torch to make the results reproducible.
- **Dataset Preparation**
  The dataset is split into a training set and a validation set. In nearly all experiments the separation is done by subject ID to inspect whether a model trained on a few subjects can make predictions on a new subject. It would also be possible to separate by sessions as done by Schirrmeister et al[31].
- **Training**
  A model is trained and validated with its corresponding hyperparameters (learning rate, weight decay, dropout, etc.) which were determined beforehand in a hyperparameter analysis for each model. As optimizer AdamW was chosen as a variant of stochastic gradient descent that is aimed towards high-dimensional parameter spaces which is optimal for the large number of parameters of a ConvNet[37].

3.5. Metrics

The performance of the experiments is measured by the most widely used metrics in classification problems to ensure great comparability. To calculate those metrics a confusion matrix is calculated which compares the predictions to the ground-truth labels. (see figure 3.6).

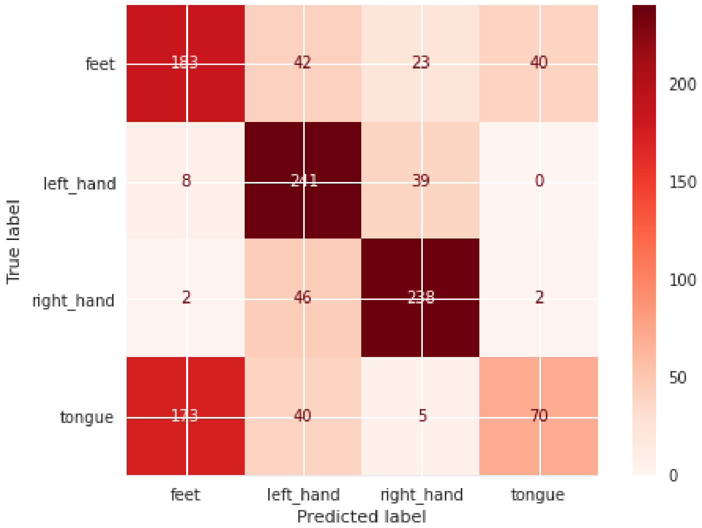

Figure 3.6.: Sample Confusion Matrix
Each row represents instances of an actual (true) class and each column the instances of a predicted class. Each actual class has 288 instances in this example and the darker a square is the more instances from that actual class have been predicted with that label. The true positives correspond to the values in the diagonal from top left to bottom right.

The metrics are calculated by the following formulas where c denotes the number of classes, the M index represents macro-averaging (equal averaging weight to different classes), tp_i stand for "true positive of class i", fn_i for "false negative of class i", fp_i for "false positive of class i" and tn_i for "true negative of class i":

- **Accuracy**
  The ratio of correctly predicted trials to the total trials. It is the most intuitive performance measure, gives a fast overview and is used in most research. Accuracy can be problematic though with unbalanced classes (one true class has more instances than another) and also it assigns equal cost to false positives and false negatives whereas depending on the application mostly one is preferred over the other[38].

$$Accuracy = \frac{\sum_{i=1}^{c} \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{c} \tag{3.1}$$

- **Precision**
  The ratio of correctly predicted trials to the total predicted positive trials. It answers the question of how many as feet movement predicted trials were actually correct. A high precision implies a low false positive rate.

$$Precision_M = \frac{\sum_{i=1}^{c} \frac{tp_i}{tp_i+fp_i}}{c} \tag{3.2}$$

- **Recall**
  The ratio of correctly predicted trials to the all trials in the actual class. It answers the question of how many of the actual feet movements were correctly labeled as that. A high recall implies a low false negative rate.

$$Recall_M = \frac{\sum_{i=1}^{c} \frac{tp_i}{tp_i + fn_i}}{c} \tag{3.3}$$

- **F1-Score**
  The harmonic mean between precision and recall. The F1-Score delivers a combined value about precision and recall that punishes extreme values by using the harmonic mean instead of the average. By maximizing the F1-Score a model is found with the optimal balance of precision and recall.

$$F1 - Score = \frac{2 * Precision_M * Recall_M}{Precision_M + Recall_M} \tag{3.4}$$

Since the datasets are evenly distributed regarding the class instances accuracy is eligible as the main metric and will be used for it also leads to a high comparability with other research where it is primarily used. It should be mentioned though that the accuracy doesn't give the whole picture of the performance of a model. Depending on the BCI application other metrics can become more important. For example a BCI-controlled machine completing sensitive tasks shouldn't move when no movement was thought of (issued via the BCI) and thus the false positive rate should be as low as possible. Therefore in such a case the precision should be considered instead of accuracy. In the simple case of classifying motor imagery without any resulting action though no preference between false positives and false negatives needs to be chosen.

3.6. Visualization: Input-perturbation network-prediction correlation map

ConvNets are hard to interpret since a trained model is defined by a huge sets of weights that can't reasonably be made sense of by a human. It is possible though to look at the ConvNet as a black box, give it a certain input and observe the output. By perturbing the original trial amplitudes, changes in the resulting predictions can be registered. By correlating the amplitude changes with the prediction changes a scalp map can be created for each class indicating where amplitude changes registered on the scalp surface have the most impact on the corresponding label. These scalp maps will be called correlation maps from now on. A sample correlation map showing a clear left-right separation for left and right hand movement imagination can be seen in figure 3.7. This is assumed to be correct due to the hand motor function left-right separation in the brain. A detailed description of how the correlation is calculated is formulated by Schirrmeister et al.[31].

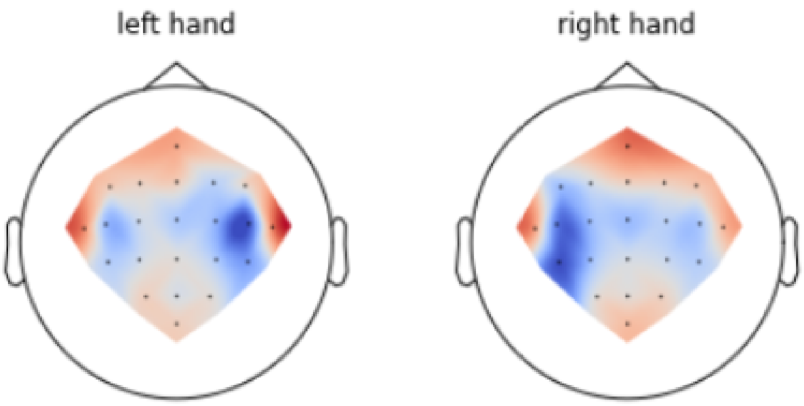

Figure 3.7.: Sample Correlation Map
The red and blue denote a positive resp. negative correlation between amplitude perturbations of the EEG signal and the network prediction in a specific frequency band (here: alpha).

4. Experiments

Three ConvNet models have been established to be used and four MI-EEG datasets. This results in twelve model-dataset combinations which were all separately trained for inter- and intra-subject classification. To get the data for the inter-subject scores, for each model-dataset combination several models were trained, one for every distinct partition, whereas a partition is made of two subsets, such that one of the subsets has a cardinality of one (e.g. [1,2],[3], [1,3],[2], and [2,3],[1] for three total subjects). Of those subsets the big one was used for training and the one with only one subject for validation. This results in s trained models per model-dataset combination where s denotes the number of subjects in the corresponding dataset. Finally the metrics of that model were then averaged to produce a more general result and reduce outlying results. For the intra-subject results the models were trained on predefined splits in the dataset, i.e runs or sessions depending on the dataset.

4.1. Model Comparison

4.1.1. Deep4Net is the best model for inter- and intra-subject classification

When comparing the models Deep4Net, ShallowFBCSPNet and EEGNetv4 over all EEG datasets, it is observable that Deep4Net has the highest average metric scores for intra- and inter-subject classification. In figure 4.1 it can be seen that Deep4Net has the best average accuracy over all datasets in the inter-subject setting and in figure 4.2 also for intra-subject classification.

The accuracy of all models trained on data from the BNCI2014001 dataset is not directly comparable to the models trained with the other datasets because they contain different amounts of classes. A 4-class classifier cannot be directly compared to a binary one since the same accuracy has a very different meaning for the two. While a 4-class classifier reaching 50% accuracy isn't half bad since he classifies every second instance correctly, a binary classifer with 50% is as bad as it gets because even a completely random classifier would reach that accuracy with only two classes. Therefore for this comparison the accuracies of all models trained on the BNCI2014001 dataset were "normalized" with equation 4.1:

$$acc_{normalizedTo2} = \frac{1 - acc_{random_2}}{1 - acc_{random_1}} * (acc_{regularIn1} - acc_{random_1}) + acc_{random_2} \tag{4.1}$$

The equation normalizes the interval of the 4-class classifier from 25% (acc_{random_1}) to 100% to the same interval as a binary classifier, from 50% (acc_{random_2}) to 100%. This is by no means a perfect comparison and should not be used in general but it gives some sense of an average performance of the considered models.

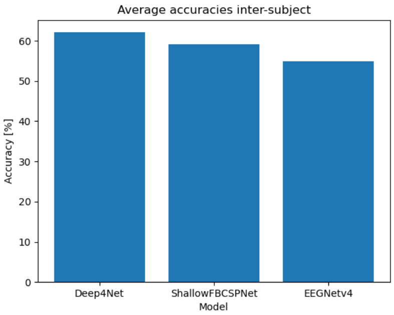

Figure 4.1.: Average accuracy over all Datasets for inter-subject classification

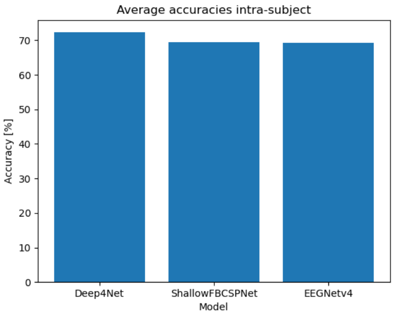

Figure 4.2.: Average accuracy over all Datasets for intra-subject classification

For inter-subject classification in figure 4.3 and for intra-subject classification in figure 4.4 the respective accuracies for each dataset are visible per model. BNCI2014001 accuracies weren't "normalized" here.

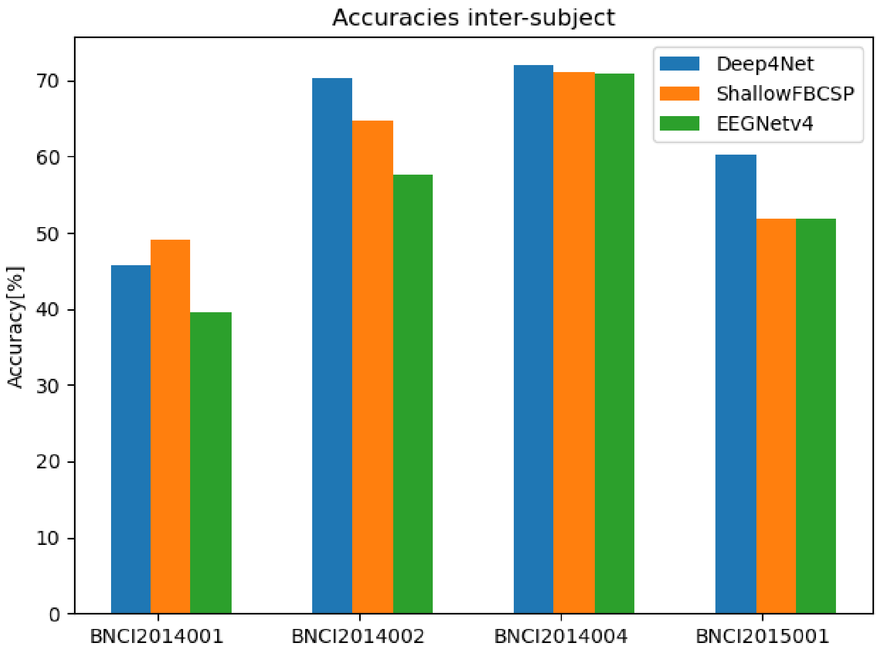

Figure 4.3.: Accuracies for each Dataset for inter-subject classification

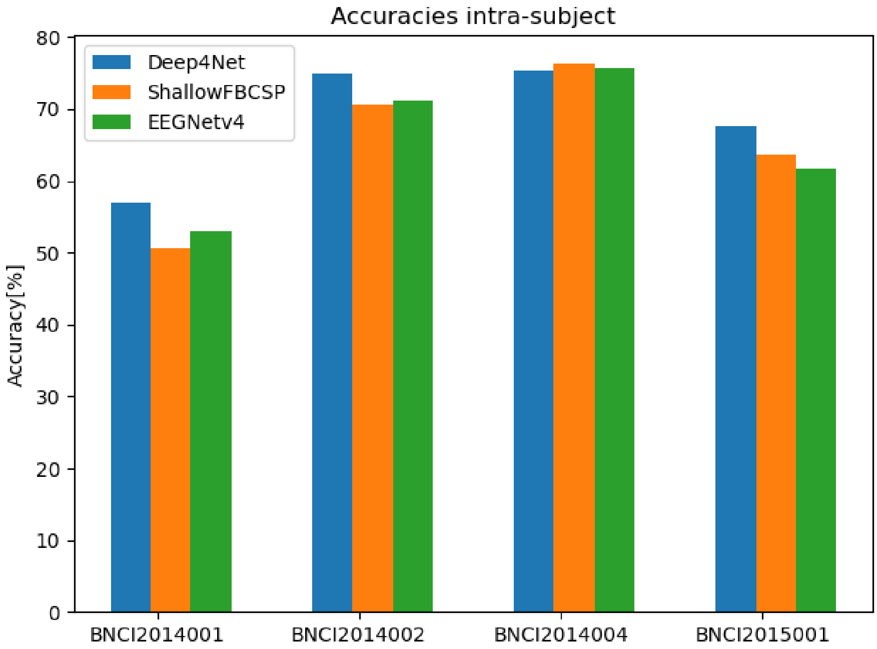

Figure 4.4.: Accuracies for each Dataset for intra-subject classification

More graphs and results with the other metrics can be found in the appendix A.2.

4.1.2. The ConvNets on average have a 38.7% performance drop for inter-subject classification compared to intra-subject classification

The performance drop $perf_{drop}$ of a model from intra-subject to inter-subject classification is calculated for each model-dataset combination with formula 4.2. acc_{inter} and acc_{intra} refer to the averaged inter- and intra-subject accuracies for each model-dataset combination and acc_{random} refers to the accuracy that a random classifier would reach for that dataset (e.g. 50% for binary and 25% for 4-class).

$$perf_{drop} = \frac{acc_{intra} - acc_{random} - (acc_{inter} - acc_{random})}{acc_{intra} - acc_{random}} = \frac{acc_{intra} - acc_{inter}}{acc_{intra} - acc_{random}} \tag{4.2}$$

This formula calculates the difference in accuracy to a random classifier for the chosen dataset (e.g. an inter-subject classifier with the same accuracy as a random classifier would lead to a performance drop of 100%).

In table 4.1 the performance drop of each model and each dataset is displayed. It can be seen that the performance drop is quite considerable for all models and EEGNetv4 performs a lot worse compared to the other two models.

	BNCI2014001	BNCI2014002	BNCI2014004	BNCI2015001	Average
Deep4Net	35.23%	18.40%	13.15%	41.94%	27.18%
ShallowFBCSPNet	5.98%	28.54%	20.29%	85.95%	35.22%
EEGNetv4	48.40%	63.07%	19.13%	83.51%	53.76%
				Average:	38.70%

Table 4.1.: Performance drop for all model-dataset combinations

4.1.3. The ConvNet models produce similar correlation maps

Figure 4.5 illustrates that the ConvNet models Deep4Net, ShallowFBCSPNet and EEGNetv4 produce similar correlation maps. This figure shows four correlation maps (one for each class) of subject 1 in the BNCI2014001 dataset for all ConvNet models over the alpha frequency band. Other subjects and frequency bands show similar results.

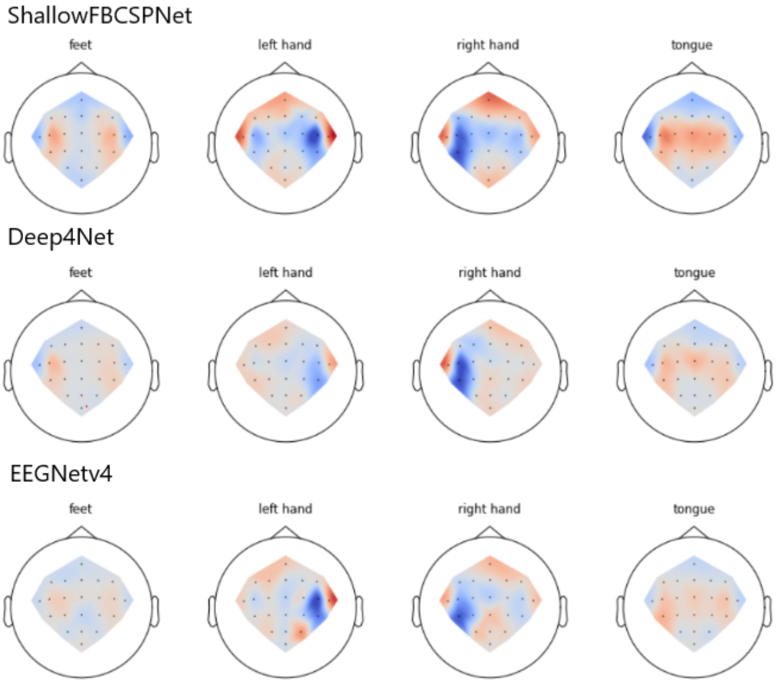

Figure 4.5.: Correlation maps of the alpha frequency band for subject 1 from BNCI2014001

4.2. Subject Comparison

4.2.1. Subjects with underwhelming classification accuracy have atypical correlation maps

Figure 4.6 shows a big accuracy variance between the suspects. While subjects from a single dataset sometimes form several distinct clusters (red and yellow datasets in figure 4.6) overall this does not seem to happen when comparing all subjects together.

When comparing the correlation maps from one of the bad performing subjects (see figure 4.7) to a well performing subject (see figure 4.8) it can be seen that the correlation maps of the bad performing subject are atypical compared to the better performing subject. For example in figure 4.8 it is visible that there is a strong negative correlation of the left brain hemisphere for right hand MI and a strong negative correlation in the right hemisphere respectively for left hand MI which is not detectable on the bad subject in figure 4.7. This also applies to other badly performing subjects (accuracy < 60 %).

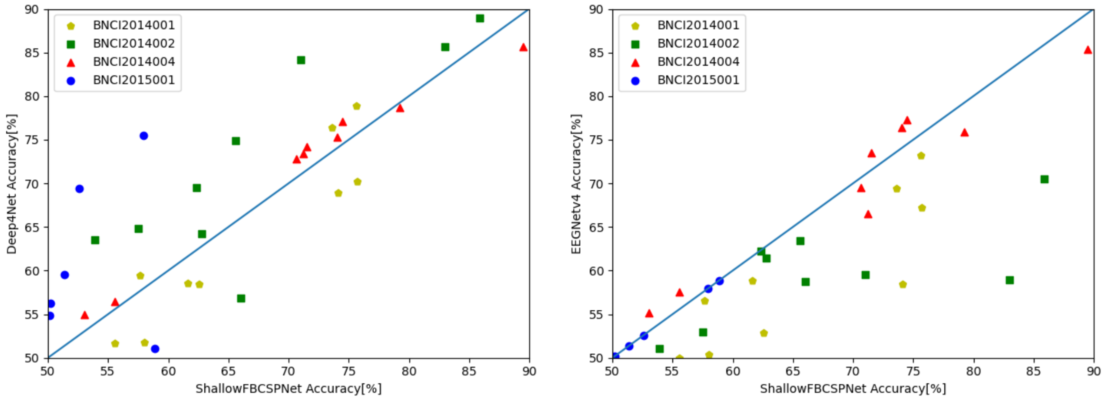

Figure 4.6.: Scatterplot of inter-subject accuracies
Each point denotes the accuracy achieved by validating a subject on a trained inter-subject model. The accuracies are color-coded by which dataset they correspond to. The accuracies of dataset BNCI2014001 were "normalized" for comparison according to equation 4.1.

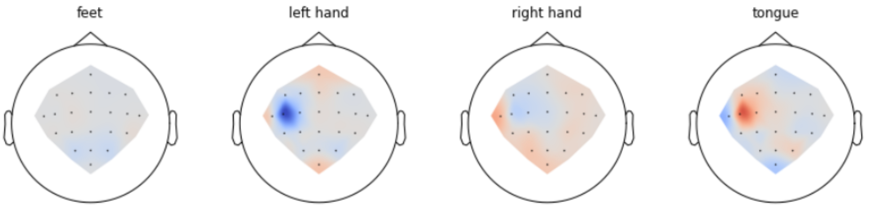

Figure 4.7.: Correlation map of a bad performing subject (subject 5 from BNCI2014001)

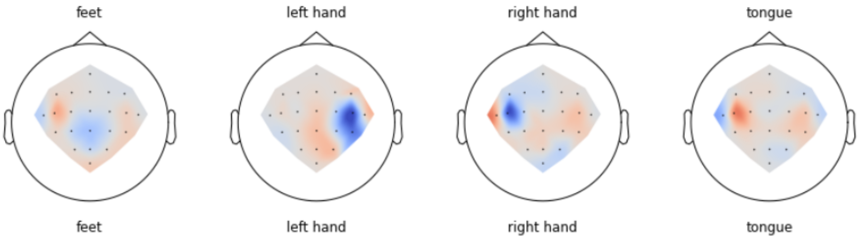

Figure 4.8.: Correlation map for a well performing subject (subject 3 from BNCI2014001)

4.2.2. Confusion matrices of badly performing subjects overfocus on a subset of classes

Figure 4.9 shows a confusion matrix of one of the worst performing subjects in the given datasets. It can be seen that the classifier completely ignores two classes and evenly distributes the guesses among these classes which results in a similar accuracy as just guessing. As opposed to this figure 4.10 shows the confusion matrix of one of the best performing subjects where most values lie on the diagonal which denotes correctly predicted instances.

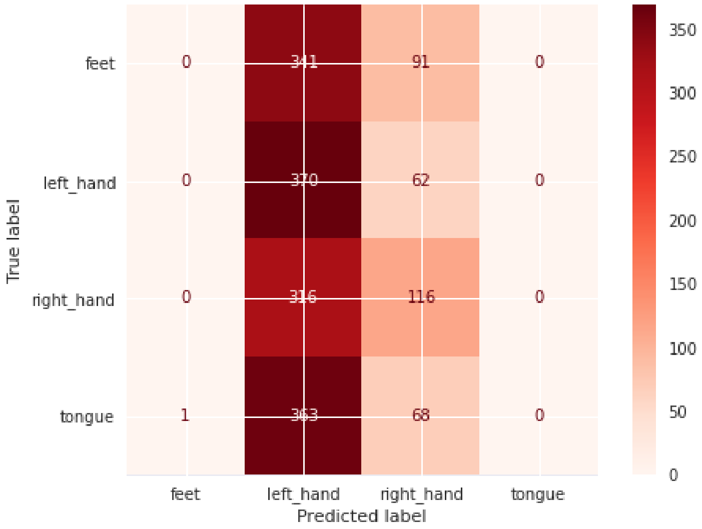

Figure 4.9.: Confusion map of a bad performing subject (subject 5 from BNCI2014001)

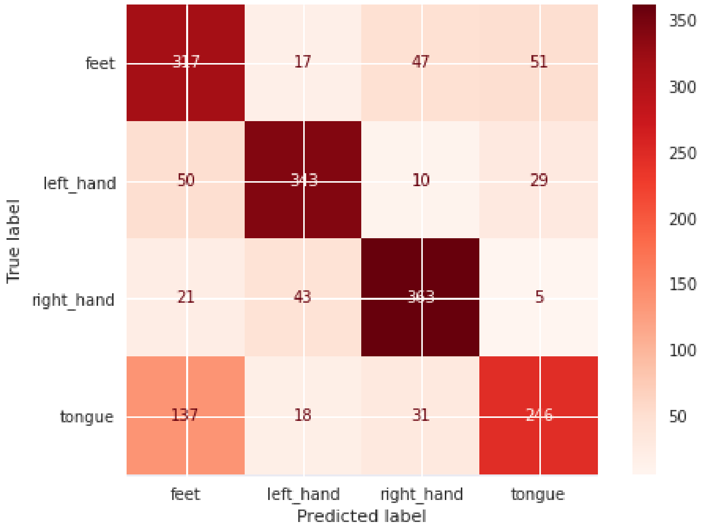

Figure 4.10.: Confusion map of a well performing subject (subject 3 from BNCI2014001)

5. Discussion and Outlook

5.1. Critique

Before we analyze our results and make assumptions it is very important to assess the limitations and relevancy of our results and to keep those in mind. The most relevant critique is that we did not reach the performances which were reported in the respective research papers of the models. This underperformance of the models, compared to other literature[31] [34] can be attributed to our lack of experience in training DL-Models, limited time frame, and missing knowledge about the used hyperparameters for each model. It is also important to note that the EEGNetv4 implementation has not been checked by the original author and is only an implementation based on the respective paper[39]. Because of this, the achieved accuracy scores should only be viewed as an indication and not as an absolute metric.

5.2. Model Comparison

Contrary to our expectation that a domain specialized ConvNet like EEGNetv4 should be the best performing ConvNet, if it is compared with more general CNN architecture, it was outperformed by Deep4Net. We could observe that Deep4Net on average performs better than the specialized EEGNetv4 on both inter- and intra-subject classification. This could be attributed to the fact that Deep4Net has more convolutional layer or that the AdamW optimizer used to train, might no be the best fit for EEGNetv4, further possible general reasons were stated in section 5.1. Because of that, it would be interesting to replicate these experiments with fine-tuned hyperparameters and in general better optimized models.

The most surprising experiment was the performance drops in result 4.1.2, it is observable that there is no guarantee that a model that performs better in intra-subject classification, also needs to perform well in inter-subject classification. For example the ShallowFBCSPNet was better than Deep4Net on the BNCI2014001 dataset in the inter-subject classification, even if Deep4Net was better on the same dataset in the intra-subject classification on the BNCI2014001 dataset. Because of our limited results which are only of indicative nature, we can not make any assumption why this behavior is happening. It would be very interesting to further analyze this behavior with differently trained models, different model types, and more/bigger datasets.

In result 4.1.3 we can see that the CNN models generated similar correlation map and if they performed well, the correlation maps are interpretable, meaning that the classification process of ConvNets in EEG-Decoding can be interpreted and evaluated which in general is a problematic topic in DL. This property could further the understanding of human brains and it speaks for the usage of ConvNets in EEG decoding.

5.3. Subjects

In result 4.2.1 it is observable that the accuracies for each subject can vary very strongly and we see that the models could not find the expected features in the correlation map and in result 4.2.2 we see that the classifier with bad performance for a subject focus on a subset of classes. This could be because of noise like body movement, a chaotic mindstate leading to difficult feature extraction or bad regularization of our models. It also could be a low motivation of the subjects which greatly impacts the performance[40].

Because some models only consider a subset of classes for some subjects as visible in the corresponding confusion matrices in result 4.2.2, we propose to introduce a penalty into the loss function for datasets with equal class distribution so that it punishes uneven classification distributions i.e overfocus on a subset of classes.

5.4. Conclusion

The results of this project are not of ideal validity. For the reasons mentioned in section 5.1 during the experiments a performance was reached that isn't comparable to other research and therefore not very reliable. A thorough hyperparameter analysis would be needed to redo the large experiments to improve their performance for which we are lacking in knowledge to determine on what to focus. But even though that the results were not very precise they still show a trend that points in the right direction.

Out of our three research objectives we were able two examine the first two, comparison between inter- and intra-subject and visualization, but due to time constraints we had to scrap the question concerning transfer learning.

Inter- vs. intra-subject

To apply EEG decoding with ConvNets there is still a lot of room for optimization shown by the considerable performance drop that was detected when validating on a new subject (inter-subject) in contrast to validating on new data from a subject that was already trained on (intra-subject).

Visualization

Visualizations by input-output network-prediction correlation maps can be used to interpret the resulting features roughly but for a detailed interpretation additional knowledge would be needed. Also we hoped to visualize the models in-between the layers which proved to be more challenging and was not achievable in the end.

5.5. Follow Up

In a next step it would be useful to expand on the two research objectives we have covered. Extending the comparison between inter- and intra-subject classification to more subjects and fine-tuning the hyperparameters or to take a look inside of the models and make sense of each layers output. Better trained models could also be easier to interpret.

6. Verzeichnisse

Bibliography

[1] A. Keen (2015), "Mr. Lucian in suburbia: links between the True History and The First Men in the Moon." Classical Traditions in Science Fiction, pp. 105-120

[2] M. Shelley (1818), "Frankenstein, or, The Modern Prometheus"

[3] V. Straebel and W. Thoben (2014), "Alvin Lucier's Music for Solo Performer: Experimental music beyond sonification", www.cambridge.org

[4] C. Zickler, et al. (2009), "BCI applications for people with disabilities: defining user needs and user requirements." Assistive technology from adapted equipment to inclusive environments, AAATE 25, pp. 185-189

[5] S. Abdulkader, et al. (2015), "Brain computer interfacing: Applications and challenges", Egyptian Informatics Journal, Volume 16, Issue 2, pp. 213-230, doi.org

[6] R. Mouad, et al. (2020), "MI-EEGNET: A novel Convolutional Neural Network for motor imagery classification", Journal of Neuroscience Methods, www.sciencedirect.com

[7] B. Facundo et al. (2017) Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks (Digital Image), www.researchgate.net/

[8] R. Sallehuddin et al.,(2013) Cancer Detection Using Aritifical Neural Network and Support Vector Machine: A Comparative Study (Digital Image),accessed: 19.12.2020, www.researchgate.net

[9] Author unknown, (Digital Image),accessed: 19.12.2020, https://camo.githubusercontent.com

[10] J. Patterson and A. Gibson (2017), "Deep learning: A Practitioner's Approach."

[11] Author unknown,(Digital Image),accessed: 19.12.2020, https://computersciencewiki.org/accessed: 19.12.2020,

[12] L.F. Nicolas-Alonso and J.Gomez-Gil (2012) "Brain Computer Interfaces, a Review." Sensors 2012, 12, 1211-1279,accessed: 19.12.2020,

[13] A. Cherninskyi, Wikipedia (Digital Image),accessed: 19.12.2020, commons.wikimedia.org

[14] J. Kumar and P. Bhuvaneswari (2012), "Analysis of Electroencephalography (EEG) Signals and Its Categorization - A Study", p. 2530 - 2531, reader.elsevier.com

[15] S. J. M. Smith (2005), "EEG in the Diagnosis, Classification, and Management of Patients with Epilepsy", jnnp.bmj.com

[16] N. Padfield et al. (2019), "EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges", p. 3, www.ncbi.nlm.nih.gov

[17] L.F. Nicolas-Alonso and J. Gomez-Gil (2012), "Brain Computer Interfaces, a Review", pp.1211-1279, www.mdpi.com

[18] H. Cho et al. (2018), "A Step-by-Step Tutorial for a Motor Imagery-Based BCI", www.researchgate.net

[19] M. Tangermann et al. (2012), "Review of the BCI competition IV" Frontiers in neuroscience, 6, p. 55

[20] D. Steyrl, et al. (2016), "Random forests in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier", Biomedical Engineering/Biomedizinische Technik, pp. 77-86

[21] R. Leeb, et al. (2007), "Brain-computer communication: motivation, aim, and impact of exploring a virtual apartment", IEEE Transactions on Neural Systems and Rehabilitation Engineering 15, pp. 473–482

[22] J. Faller, et al. (2012), "Autocalibration and recurrent adaptation: Towards a plug and play online ERD- BCI", IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp. 313-319

[23] B. Xu et al. (2019), "Wavelet Transform Time-Frequency Image and Convolutional Network-Based Motor Imagery EEG Classification" in IEEE Access, vol. 7, pp. 6084-6093, ieeexplore.ieee.org

[24] S. Kumar, et al. (2016), "A Deep Learning Approach for Motor Imagery EEG Signal Classification," 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), pp. 34-39, ieeexplore.ieee.org

[25] J. Kevric and A. Subasi (2017), "Comparison of signal decomposition methods in classification of EEG signals for motor-imagery BCI system" Biomedical Signal Processing and Control, Volume 31, pp. 398-406, www.sciencedirect.com

[26] M. Tangermann (2012), "Review of the BCI Competition IV", Frontiers in Neuroscience, Volume 6, www.frontiersin.org

[27] Kai Keng Ang, et al. (2008), "Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface," 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, 2008, pp. 2390-2397, (Digital Image), www.researchgate.net/accessed: 19.12.2020,

[28] Kai Keng Ang, et al. (2012), "Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b", Frontiers in Neuroscience, Volume 6, www.frontiersin.org

[29] X. Zhang, et al. (2020), "A Survey on Deep Learning-based Non-Invasive Brain Signals:Recent Advances and New Frontiers", p. 21, arxiv.org

[30] M. Miao, et al. (2020), "Spatial-Frequency Feature Learning and Classification of MotorImagery EEG Based on Deep Convolution Neural Network", p. 2 downloads.hindawi.com

[31] R. Schirrmeister, et al. (2017), "Deep Learning With Convolutional Neural Networks for EEG Decoding and Visualization", Human Brain Mapping, pp. 5391-5420, onlinelibrary.wiley.com

[32] Y. Tabar and U. Halici (2017), "A novel deep learning approach for classification of EEG motor imagery signals", p. 2, iopscience.iop.org

[33] P. Bashivan, et al. (2016), "Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks", arxiv.org

[34] V. Lawhern, et al. (2018), "EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces", pp. 6-7, www.researchgate.net

[35] R. Schirrmeister, et al. (2017), "Supplementary for: Deep learning with convolutional neural networks forEEG decoding and visualization", onlinelibrary.wiley.com

[36] S. Ioffe and C. Szegedy (2015), "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arxiv.org

[37] I. Loshchilov and F. Hutter (2019), "Decoupled Weight Decay Regularization", arxiv.org

[38] M. Sokolova and G. Lapalme (2009), "A systematic analysis of performance measures for classification tasks", p. 430, rali.iro.umontreal.ca

[39] Braindecode developers (2020), "braindecode.models.EEGNetv4 - Braindecode 0.5 documentation", accessed: 19.12.2020, braindecode.org

[40] R. Leeb, et al. "Brain–Computer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment", ieeexplore.ieee.org

[41] Braindecode developers (2020), "Welcome to Braindecode", accessed: 19.12.2020, braindecode.org

[42] Jupyter Team (2015), "The Jupyter Notebook - Jupyter Notebook 6.1.5 documentation", accessed: 19.12.2020, jupyter-notebook.readthedocs.io

[43] GitHub Inc. (2020), "GitHub repository buergmax/PA_from_human_to_artificial_neural_networks", accessed: 19.12.2020, github.zhaw.ch

[44] Institute of Applied Information Technology (2020), "InIT GPU Infrastructure", accessed: 19.12.2020, github.zhaw.ch

[45] Docker Inc. (2020), "Docker Desktop overview", accessed: 19.12.2020, docs.docker.com/

[46] Sylabs Inc. (2020), "Singularity User Guide", accessed: 19.12.2020, sylabs.io

[47] Alexandre Barachant (2018), "Mother of all BCI Benchmark - moabb documentation", accessed: 19.12.2020, moabb.neurotechx.com

[48] V. Jayaram and A. Barachant, et al. (2018), "MOABB: trustworthy algorithm benchmarking for BCIs", Journal of Neural Engineering, Volume 15, Number 6, iopscience.iop.org

List of Figures

2.1. Artificial Neural Network[7] . . . 3
2.2. Neuron inside a ANN[8] . . . 4
2.3. Convolutional Neural Network[9] . . . 4
2.4. Convolution Illustration[10] . . . 5
2.5. Max-Pooling Illustration[11] . . . 5
2.6. EEG Sensors on Subject . . . 6
2.7. EEG Data . . . 6
2.8. Architecture of the filter bank common spatial pattern (FBCSP) algorithm for the training and evaluation phases.[27] . . . 7

3.1. Trialwise Decoding . . . 8
3.2. Cropped Decoding . . . 8
3.3. Overall visualization of the EEGNet architecture . . . 9
3.4. Architecture of the shallow ConvNet designed by Schirrmeister et al . . . 10
3.5. Architecture of the Deep4 ConvNet designed by Schirrmeister et al . . . 11
3.6. Sample Confusion Matrix . . . 14
3.7. Sample Correlation Map . . . 15

4.1. Average accuracy over all Datasets for inter-subject classification . . . 16
4.2. Average accuracy over all Datasets for intra-subject classification . . . 16
4.3. Accuracies for each Dataset for inter-subject classification . . . 17
4.4. Accuracies for each Dataset for intra-subject classification . . . 17
4.5. Correlation maps of the alpha frequency band for subject 1 from BNCI2014001 . . . 18
4.6. Scatterplot of inter-subject accuracies . . . 19
4.7. Correlation map of a bad performing subject (subject 5 from BNCI2014001) . . . 19
4.8. Correlation map for a well performing subject (subject 3 from BNCI2014001) . . . 19
4.9. Confusion map of a bad performing subject (subject 5 from BNCI2014001) . . . 20
4.10. Confusion map of a well performing subject (subject 3 from BNCI2014001) . . . 20

A.1. Averaged recall inter-subject classification . . . III
A.2. Averaged recall intra-subject classification . . . III
A.3. Averaged f1 inter-subject classification . . . IV
A.4. Averaged f1 intra-subject classification . . . IV
A.5. Averaged precision inter-subject classification . . . V
A.6. Averaged precision intra-subject classification . . . V

List of Tables

4.1. Performance drop for all model-dataset combinations 18

A. Appendix

A.1. Environment Setup

This projects code can either be run with a local installation of the Braindecode toolbox[41] and Juypter Notebook[42] (installation links are given with citations) or in a docker container generated with the Dockerfile in A.1 (use is described in section A.1.3). The code is found on GitHub[43].

The following sections describe the used environment for the experiments.

A.1.1. GPU cluster

The use of a High Performance Computing (HPC) system is recommended. This allows the neural networks to be trained with faster computing units. A GPU cluster from the ZHAW is available for this purpose. This is operated and provided by the 'Institute of Applied Information Technology' (InIT). The InIT GPU Cluster is made up of four nodes with each consisting of two Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz with 12 Cores and eight Nvidia TITAN Xp graphics cards[44].

A.1.2. Software

Docker

Docker is an effective tool to build, run and share containerized applications. This means an application becomes more portable, scalable and secure by using Docker. The program can be run on any OS supported by Docker, can be distributed across multiple machines and is isolated from the underlying system[45]. For this project Docker was used to keep the test environment constant for all experiments. The Docker image is created by a Dockerfile containing the installation of the Braindecode toolbox (see A.1).

Singularity

Singularity is a container system, similar to Docker, with the difference that Singularity creates containers, which can be started on a high performance computer (HPC) without root privileges. Docker would require root access which is undesirable from a security perspective on a HPC system. Singularity can convert a Docker image to a Singularity image and run the container without elevated privileges[46].

Jupyter Notebook

Jupyter Notebook is a browser-based tool to enrich python scripts with live code, explanatory text and visual elements. This tool provides a simple GUI Access to the project code and allows code to be executed piece by piece to avoid the retraining of a whole model when only working with the result[42].

MOABB - Mother of all BCI Benchmark

MOABB is an open-source software suite that provides a simple access to different BCI datasets and a consistent interface to apply state-of-the-art decoding algorithms to this data. Braindecode uses MOABB mainly to load freely available EEG datasets[47][48].

A.1.3. Running with Docker

Build the image with:

```
docker build -t <image_name> -f <path_to_dockerfile> .
```

Create and run a container after image creation with:

```
docker run -it --name <container_name> <image_name>
```

Start and stop a container with:

```
docker start/stop <container_name>
```

Access bash shell inside a running container:

```
docker exec -it <container_name> /bin/bash
```

A.1.4. Dockerfile

```
FROM python:3.8-slim-buster

WORKDIR /usr/src/app

# Installation for braindecode
RUN pip install torch && \
pip install download && \
python -c "from download import download; download(’https://raw.githubusercontent.com/NeuroTechX/moabb/master/requirements.txt’, ’requirements.txt’, replace=True)" && \
pip install -r requirements.txt && \
rm requirements.txt && \
pip install -U https://github.com/NeuroTechX/moabb/archive/master.zip && \
pip install braindecode && \
pip install jupyterlab

CMD [ "python"]
```

Listing A.1: The Dockerfile used to generate a docker image to run the projects code.

A.2. Model Comparison - Other Metrics

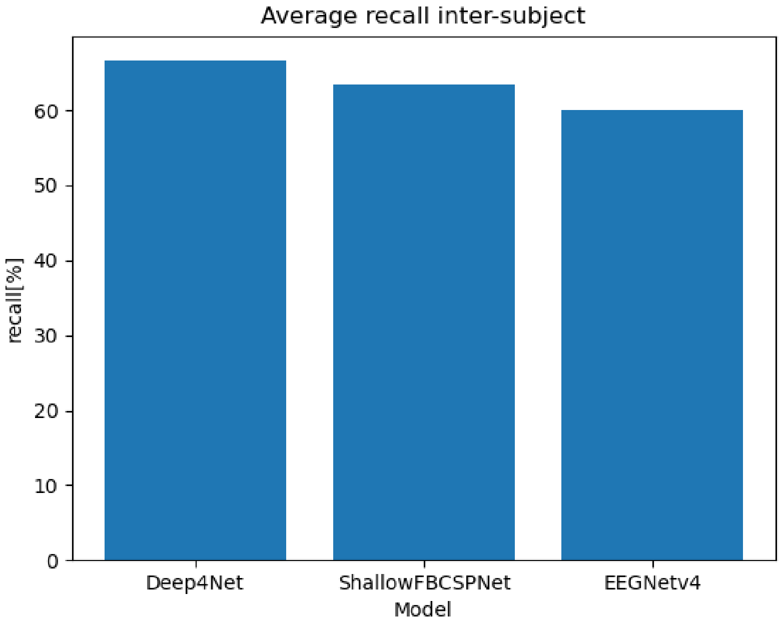

Figure A.1.: Averaged recall inter-subject classification

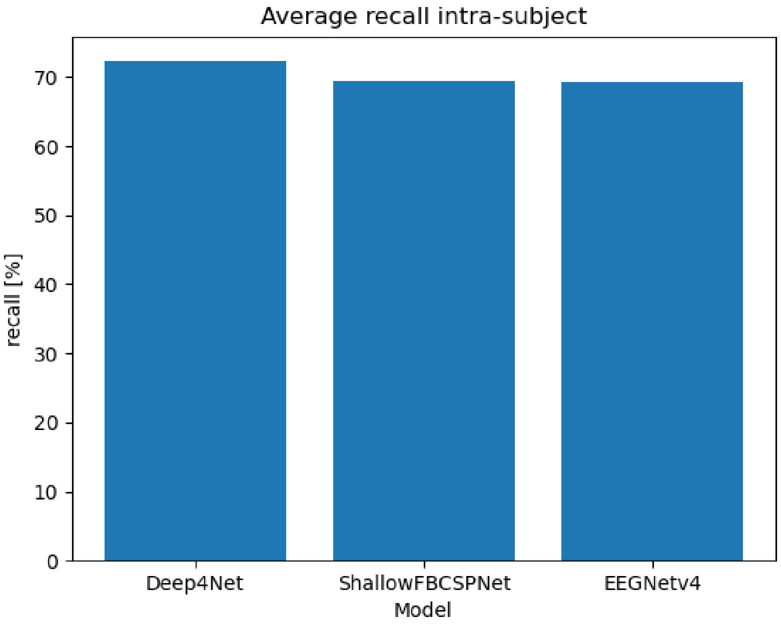

Figure A.2.: Averaged recall intra-subject classification

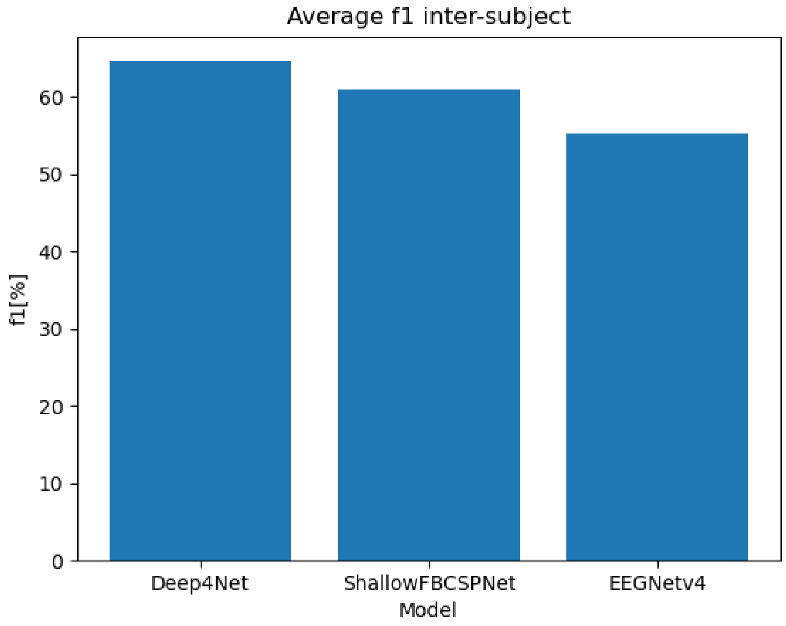

Figure A.3.: Averaged f1 inter-subject classification

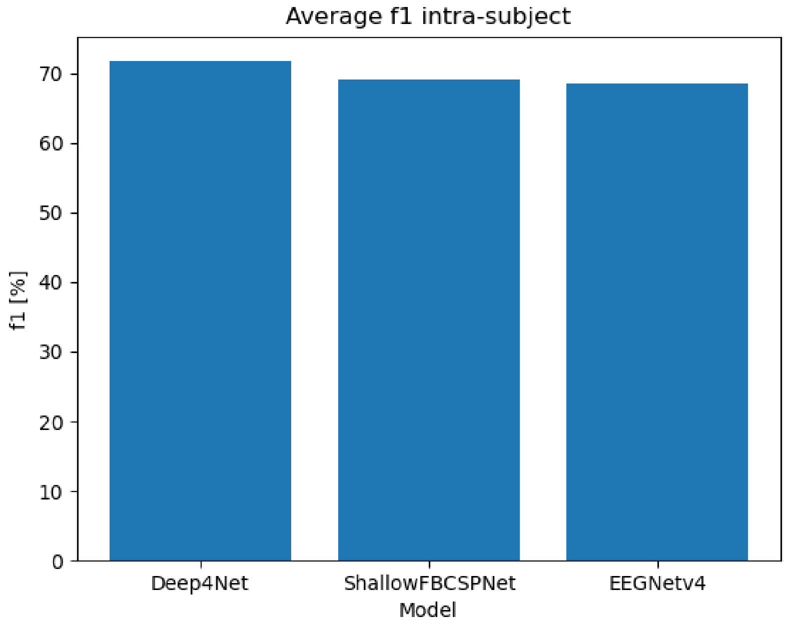

Figure A.4.: Averaged f1 intra-subject classification

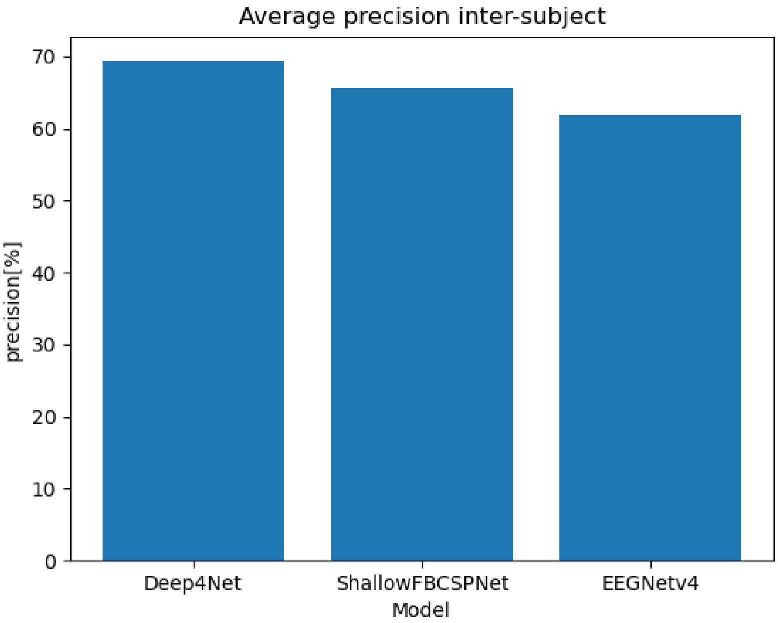

Figure A.5.: Averaged precision inter-subject classification

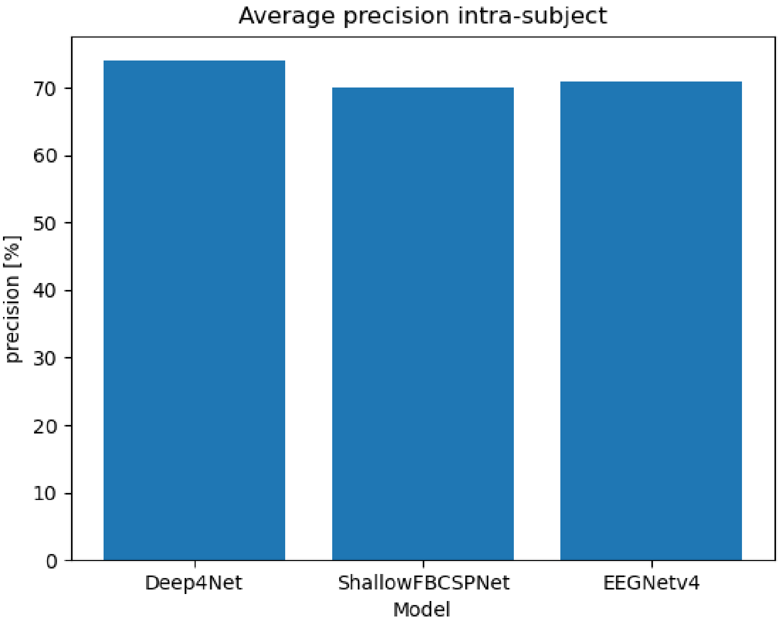

Figure A.6.: Averaged precision intra-subject classification