



NTNU – Trondheim
Norwegian University of
Science and Technology

Designing an EEG based communication system for patients with Locked-in Syndrome

Tobias Treider Moe

December 2021

SPECIALIZATION PROJECT

Department of Engineering Cybernetics

Norwegian University of Science and Technology

Supervisor 1: Marta Molinas

Supervisor 2: Luis Alfredo Moctezuma

Abstract

This work tests a possible Brain-Computer Interface (BCI) design that can be used for communication for persons with locked-in syndrome (LIS). Persons diagnosed with LIS are conscious and awake but trapped within their bodies, unable to move any muscle except their eyes. The tested design utilizes eye movement and color detection recognized in recorded electroencephalography (EEG) signals. The main challenge and motivation is to create an accurate and fast predicting BCI, which will not exhaust the user.

To test the proposal, an EEG dataset from 33 subjects was collected. The EEG data were collected from 8 EEG channels while the subject was exposed to two different protocols based on eye movement. Both protocols have five different classes, a rest class, and four different task classes. The first protocol utilized eye movement and color perception, while the second utilized the same as the first protocol, but also added image association.

Aiming for a real-time implementation, the problem was divided into two different challenges solved with two different models. The first model was designed to differentiate between resting-state and any other task. Once the task is identified, the second model is planned to be used, which differentiates between four classes. A state-of-the-art Convolutional Neural Network (CNN) design was applied for classification. Its hyperparameters were tried optimized with a hyperparameter search algorithm, Hyperband. Two different experiments were conducted, creating models based on individual subject data and creating models with cross-subject data.

The highest accuracy created with cross-subject data were 80.6% on a resting-state vs task model and 65.1% on a 4-class model. So the experimental results obtained with the dataset shows that the BCI design where possible to implement. However, more advanced research and development will be necessary before the BCI can be implemented into a real-time application for a person diagnosed with LIS.

Contents

Abstract	i
1 Introduction	2
1.1 Problem description	3
1.2 Approach	3
1.3 Report structure	3
2 Background	4
2.1 Locked-in Syndrome	4
2.2 The human brain	4
2.2.1 Brain activity	4
2.2.2 Color vision and occipital lobe	5
2.2.3 Eye movement	6
2.3 Electroencephalography	6
2.3.1 EEG frequency bands	7
2.3.2 Electrode placement	7
2.3.3 Artifacts	7
2.3.4 Recording protocol	8
2.4 Multi-class classification with Convolutional Neural Network	8
2.4.1 Hyperparameter optimization with Hyperband	9
3 Literature review	10
3.1 Previous BCIs for LIS patients	10
3.2 EEG classification with CNNs	11
4 Materials and methods	13
4.1 Hardware tools	13
4.2 Electrode placement	14
4.3 EEG data collection	15
4.3.1 Protocols	15
4.3.2 Metadata of the dataset	16
4.4 Pre-Processing	17
4.5 EEGNet architecture	19
4.6 Methods for evaluation	20

5	Experimental results	22
5.1	Evaluation of subject performance	22
5.1.1	Individual models based on P1 data	23
5.1.2	Individual models based on P2 data	23
5.1.3	Models based on P1 and P2 data	23
5.2	Evaluation of subject performance with a sample length of 1 seconds	27
5.2.1	Models based on P1 data	27
5.2.2	Models based on P2 data	27
5.2.3	Models based on P1 and P2 data	27
5.3	Evaluating individual P2 data on individual subject P1 models	31
5.4	Cross-subject evaluation	31
5.4.1	Models trained with P1 data	31
5.4.2	Models trained with P2 data	32
5.4.3	Models trained with P1 and P2 data	33
5.5	Cross-subject evaluation with a selection of subjects	34
5.5.1	Models trained with P1 data	35
5.5.2	Models trained with P2 data	36
5.5.3	Models trained with P1 and P2 data	37
6	Discussion and further work	39
6.1	Discussion	39
6.1.1	Protocols and data acquisition	39
6.1.2	Classifiers	41
6.2	Further work	43
	References	45

Chapter 1

Introduction

Persons diagnosed locked-in syndrome (LIS) are conscious and awake, but trapped within their bodies, with little to no possibilities to communicate. They have *quadriplegia*, a partial or total loss of control over all four limbs and torso, and *anarthria*, a loss of control in the muscles used for speaking [1]. There are several variants of LIS, and in the most widespread version, the muscles that control vertical eye movement and blinking is still intact.

Adding to their condition, persons with LIS also face social barriers through social exclusion, stigmatization, and frequently being underestimated. In other words, they are being 'locked out' in addition to being 'locked in'. A study [2] discusses that when persons are diagnosed with LIS, other people do not treat them as human beings anymore. As if muteness and paralysis mean being mentally impaired or devoid of feelings or preferences. The consequence is that people rarely speak directly to them. They either speak around, over or about them. In addition, there is a failure to treat LIS persons as aware and conscious beings with interests and preferences worthy taking into account; which could be a violation of the United Nations Convention on the Rights of Persons with Disabilities [2]. The failure comes from poor understanding of the condition, making it morally critical and necessary to ensure that persons with LIS have a voice. Another study [3] has shown that a LIS-persons' ability to communicate and do activities directly influence their quality of life.

A Brain-Computer Interface (BCI) establishes a direct communication pathway between the brain, the central nervous system (CNS) and a computer, by measuring the activity and translating the information from the brain. It provides a way to communicate without relying on muscle contractions and has primarily been developed to help people with severe disabilities [4]. It measures electroencephalography (EEG) activity and records electrical signals from the brain. There are two ways to generate brain signals, active and passive. The user of the BCI can either control mental activities, or react to stimuli from the real world to actively generate signals. In comparison, passive signals monitor the users' cognitive state, e.g., drowsiness and emotional states [5]. A BCI has five stages [6]: signal acquisition, pre-processing (enhance signal-to-noise ratio), feature extraction, classification, and control interface.

A BCI used as a communication platform designed to work for persons with LIS who still have preserved eye movement, will be tested in this work. As they still have other means of communication, it

will be possible to get feedback, making this an excellent first step to test if it will work for their needs.

1.1 Problem description

This project aims to test the possibility of a new communication platform for patients with LIS. More precisely, the platform will use a BCI that processes the brain's color perception and eye movement. It is required to test different experimental configurations to verify whether it is possible to classify color perception and eye movement. The results from the test will be used as a recommendation for further real-time implementation of the communication platform.

1.2 Approach

The approach consists of two parts. The first is to design a protocol that can mimic the BCI and recording a dataset with said protocol on healthy subjects. The second part consists of classifying the dataset with a state-of-the-art neural network. The classifying speed and accuracy should be the deciding factors when measuring how good the design is, and how functional the BCI is. In addition to how exhausting the BCI is to use.

1.3 Report structure

Chapter 1 introduces the problem, and in chapter 2 presents the relevant background information about the brain, collecting signals from the brain and classification with neural networks. In chapter 3, literature of previously researched BCIs for persons with LIS and state-of-the-art neural networks used for EEG classification are briefly reviewed. The methods used to create the dataset, and the neural network and evaluation methods are presented in chapter 4. In chapter 5 the performed experiments and their corresponding results are shown. In chapter 6 the protocol and the results are discussed, and further work is suggested.

Chapter 2

Background

2.1 Locked-in Syndrome

LIS is a rare condition resulting from a brainstem lesion, which can have resulted from stroke, injury, or drug addiction. The outcome is nearly total paralysis, though cognitive functions are still operative. In addition, most can still see, hear, smell, taste, experience heat, cold, pain and pleasure [2]. However, diplopia or blurry vision can obscure their eyesight [1]. There is currently no cure available for persons diagnosed with LIS, only supportive and communicative care [7].

LIS are typically categorized into three subtypes: classical, partial and complete locked-in syndrome. A person with classical locked-in syndrome, called LIS in this work, has no mobility other than blinking or vertical eye movement. Partial locked-in syndrome is classified for persons who have preserved some motor functions, e.g., one of their legs. The most severe type is Complete Locked-in Syndrome (CLIS). Persons with this condition cannot move at all, including blinking and horizontal eye movement, making them unable to communicate [1].

2.2 The human brain

The human brain is part of the CNS together with the spinal cord, and helps with sensing, thinking, and controlling the body. The brain consists of three parts: the cerebrum, brainstem and cerebellum, visualized in fig. 2.1. The brainstem connects the spinal cord and the brain and is where LIS patients have been injured, as they no longer can transmit muscle control signals from their brain to the rest of their body.

2.2.1 Brain activity

The brain and the rest of the nervous system send signals with the help of chemical and electrical signals between neurons. Neurons consist of a nucleus, axon (the output from the neuron) and dendrites (the inputs to the neuron). A signal is sent when the dendrites' potentials sum above a threshold, around -55 mV. The potential is a depolarisation from their resting potential of around -70 mV. When the potential exceeds the threshold, the neuron will get an action potential and fire an electrical signal along the axon, which connects to a dendrite, called the synapse. The neuron will always send a signal with the same strength, though it can vary its frequency [9].

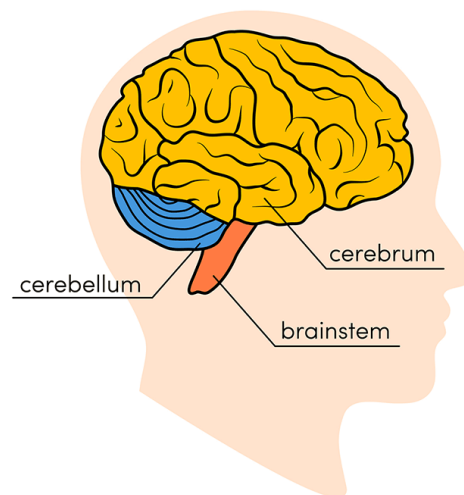
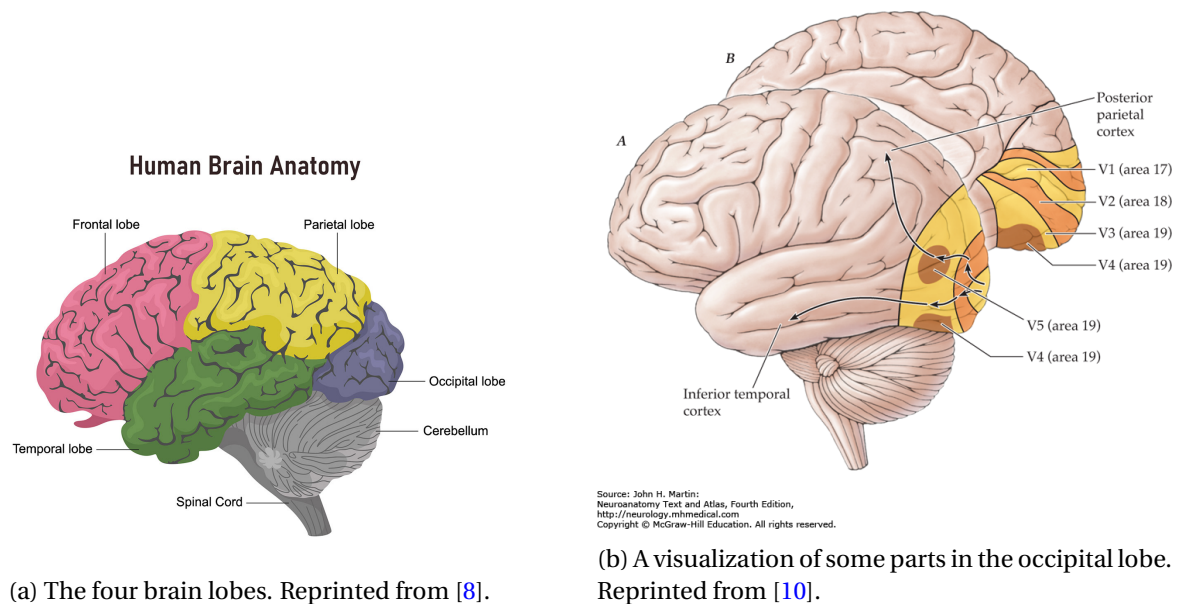


Figure 2.1: Map of the three main parts of the brain. Reprinted from [8].

2.2.2 Color vision and occipital lobe

The cerebrum divides into four sections, called lobes, where each lobe controls different functions. One of the lobes is the occipital lobe, placed in the back of the head, and is the part of the brain that is involved with vision, see fig. 2.2a.



(a) The four brain lobes. Reprinted from [8].

(b) A visualization of some parts in the occipital lobe. Reprinted from [10].

Figure 2.2: Visualization of the four brain lobes as well the main parts in the occipital lobe.

The eyes are the primary sensory organ involved with vision, which receives light waves processed in the occipital lobe. There are two different light sensory cells, cones and rods, in the eye, where the cones are directly involved in color vision. Humans have three different cones in their eyes, each with its specific pigment type, making them sensitive to different wavelengths. One responds best to light

with wavelengths in the blue range, another in the red range, and the last in the green range. Together they help to see and differentiate colors [11].

The occipital lobe consists of several parts, some of which are visualized in fig. 2.2b. The light perceived by the rods and cones is sent directly to the Primary Visual Cortex (V1) with several neurons that connect the eyes with the occipital lobe. Here the signals are being split and sent to the different regions. The color information is *mainly* being sent to Visual Area 4 (V4) [12], where it makes us perceive the colors. It is important to note that it is not only the cells in V4 that react to colors, other cells in the occipital lobe will also have a response [11].

2.2.3 Eye movement

The eyeball is placed in the orbita and has six muscles in three antagonistic pairs. These muscles have tiny muscle fibers compared to other muscles, as they perform precise movements. They are connected to the wall in orbita, and all six muscles contribute to horizontal, vertical, and rotatory motions. It is the brain stem and cerebellum that control these muscles, in addition they also coordinate the muscle contractions [11]. The muscle contractions discharge electrical activity when stimulated with a nerve pulse. Furthermore, the electrical activity from the muscles is much higher than that in the neurons, meaning that eye movement is visible when reading the EEG signals.

2.3 Electroencephalography

EEG is a method that measures the electrical activity in the brain generated by the synchronized activity of thousands of neurons. These neurons are placed perpendicular to the brain's surface, and the readings are from the summation of the postsynaptic potential [13]. The activity is measured with either intracranial electrodes (invasive method) or on the scalp's surface (non-invasive method). Electrodes will only detect a portion of the signals in the brain, and the signals will be a sum of many postsynaptic potentials in the close vicinity of the electrode. The signal from an electrode is denoted as a *channel*.

A problem with reading non-invasive EEG signals is that the measured signal can be overwhelmed by other electrical interference from the body (e.g., muscle contraction), or weakened by traversing from its origin to the electrode. It must traverse the brain, cerebrospinal fluid, meninges, skull, and skin, which have different resistance before being detected and read. The principle of differential amplification enhances the signal-to-noise ratio of the EEG signal. The method compares the "reading" electrode with a distant reference electrode, which should have an infinite distance from the source [13].

Recording EEG signals is overall very safe, both for the patient and the technologist. Still, the subject is exposed to electrical current, and the most critical injuries are skin burns, induction of seizures, or ventricular fibrillation. The sources for harmful current can be improper grounding, leakage, or double-grounding [13].

2.3.1 EEG frequency bands

The frequencies in the EEG signals separates into five different frequency bands that correspond to the subject's condition. The frequency bands can vary slightly, based on the laboratory or recording device used [14]. In addition, the literature disagrees on which band is associated with what frequencies. In table 2.1 the different bands with their frequency ranges and what they are associated with, are presented.

Frequency band	Frequency range [Hz]	Associated with
Delta	0.5 - 4	Homeostatic sleep drive
Theta	4 - 8	Associated with homeostatic sleep drive
Alpha	8 - 12	Relaxed wakefulness and drowsiness
Beta	13 - 30	Active thinking
Gamma	> 30	Cognitive states

Table 2.1: Frequency bands and their associations, values and association taken from [14].

2.3.2 Electrode placement

There exist different layouts the electrodes can be positioned on the skull after. This work will use the most acknowledged layout, the 10-20 international system, see fig. 2.3. This system represents standard electrode placement intervals of either 10 or 20 percent between the electrodes. Each electrode site has a letter and number, where the letter represents the lobe while the numbers represent the side of the brain. Odd numbers represent electrodes on the left side, and even numbers on the right side. Electrodes at the center are marked with the letter z , instead of a number.

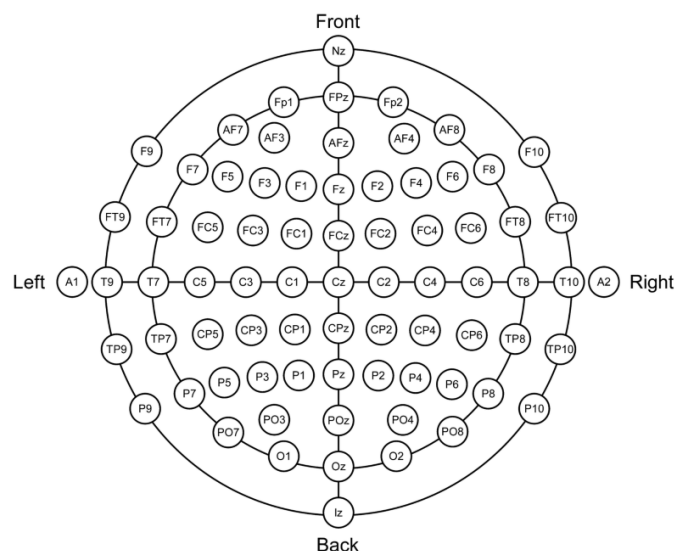


Figure 2.3: The international 10-20 system for electrode placement. Reprinted from [15].

2.3.3 Artifacts

Artifacts are electrical signals detected by the electrodes, but do not originate from the cerebrum. They can be divided into two different types: *physiological* and *non-physiological* artifacts. The most

common and researched physiological artifact is from the eye. Eye blinks, movement, and extraocular muscle activity can easily contaminate the signal. Other physiological artifacts are other muscle contractions, tongue movement, speaking, chewing, sweat, and respiration [13]. Sweat can alter the impedance of the electrodes, while respiration can introduce a rhythmic activity in the signal. Non-physiological artifacts can occur from a poor connection between electrodes and the user, power line noise (50 Hz in Europe), or unique environmental, electrical noise produced by, e.g., ventilators and feeding/infusion pumps [16]. These artifacts might not be applicable in an EEG laboratory, but may be outside such a controlled environment.

2.3.4 Recording protocol

In order to generate the active signals and record them, a recording protocol is needed. It consists of tasks given to a participant to follow in a time window while recording the brain activity. Between the tasks are the participant instructed to rest, as it is equally important to understand when a user is performing a task or not and differentiate the different tasks.

However, there are some problems with the resting state, as a stable resting state does not necessarily exist. There are, all the time, spontaneous changes in regional neural firings in the brain. As well can the spontaneous activations change the local blood flow, which can give signal oscillation [17]. Implying that the brain is never entirely at rest, so a resting state refers to when there is no goal-directed neuronal action with the integration of the external environment and when the subject is not actively engaged in sensory or cognitive processing [14].

2.4 Multi-class classification with Convolutional Neural Network

EEG-based BCIs use Machine Learning (ML) algorithms for classification. The algorithms that work best are adaptive classifiers, e.g., Riemannian geometry classifiers and adaptive support vector machine (ASVM) [18]. On the other hand, Deep Learning (DL) has not been that successful, even though it has had success in image and speech classification. DL algorithms need a large dataset to perform well, which is seldom the case of an EEG dataset. Some of the reasons behind this are cognitive tiredness, inability to continue recording, risk of epilepsy, or too expensive recording. However, DL algorithms decrease the need for manual feature extraction while acquiring state-of-the-art performance, especially true for a Convolutional Neural Network (CNN). Nevertheless, several attempts to classify EEG with CNNs have been conducted, although its poor historical results.

CNNs is built up with several *layers*, always an input and output layer, and $n \in \mathbb{N}$ *convolution layers*, called *hidden layers*. An example of the layers and architecture from a signal to a class are visualized in fig. 2.4. Only the hidden layers and functions used in this project are explained. Central in a CNN is the *convolution operation*, which is a linear operation on two functions. A *pooling layer* replaces the output at a certain location with a summary statistic of the nearby outputs, reducing the number of outputs [19]. Such a statistic can be the average of the n neighboring inputs. A *depthwise convolution* layer convolves on each channel separately. While a *separable depthwise convolution* layer first performs a depthwise convolution, then a pointwise convolution, so the channels are getting mixed. At the end of each convolutional layer is an *activation function* which defines how the weighted sum of

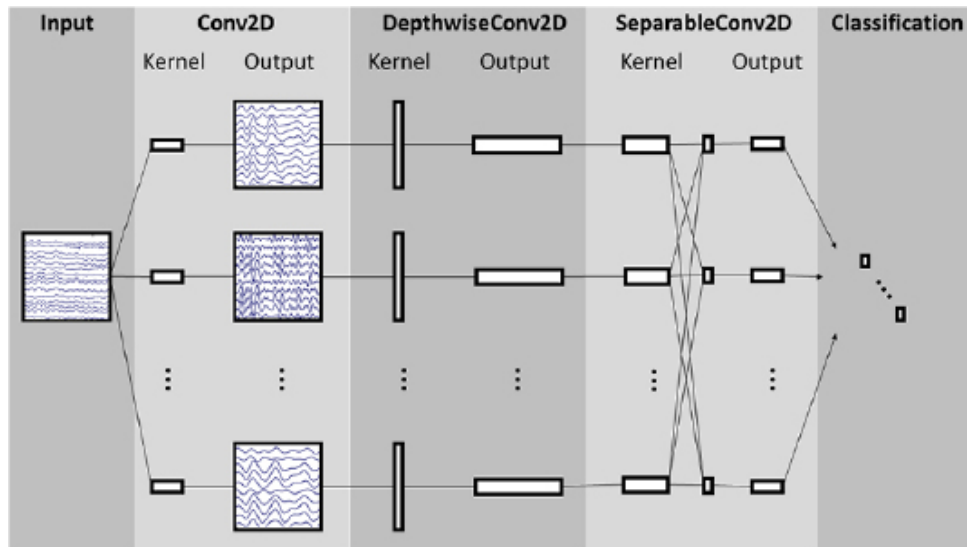


Figure 2.4: Visualization of a CNN architecture, the EEGNet, used for EEG signal classification. Reprinted from [20].

the inputs are transformed to an output. For example, it can be an Exponential Linear Unit (ELU) or softmax function.

A CNN trains over several epochs. An epoch is a term for when the network has trained on all samples in the dataset. So the repeatedly tuning of the weights with the same data can result in overfitting. Overfitting is when the model has created a very good fit to the training data, though has not found a generalized fit that can work on similar data. Two methods to combat overfitting are *dropout* and *early stopping*. A dropout layer randomly drops hidden units during training. While early stopping ends the training before intended, when it meets a criterion. A problem with early stopping is that it might end the training too early.

2.4.1 Hyperparameter optimization with Hyperband

Hyperparameters are non-trainable parameters that have to be defined prior to the training. These parameters can significantly impact the model, and several methods search automatically for the best set of hyperparameters to a dataset. These methods have all in common: The search will take a long time, as the search space might be very large. An algorithm for hyperparameter optimization is Hyperband [21]. This algorithm finds good hyperparameters relatively fast by doing adaptive resource allocation.

Hyperband begins with exploration, with an aggressive early stopping, so it does not allocate a considerable amount of resources to the hyperparameters that result in a poorly model. It will also discard configurations that resulted in such a model and not look in that "region" of the search space anymore. Configurations resulting in a good model, will convey more resources until there are no more of the pre-defined resources, and the algorithm will stop. So Hyperband intelligently allocates resources to maximize the number of possible configurations while preserving the wanted structure between the individual configurations.

Chapter 3

Literature review

3.1 Previous BCIs for LIS patients

Today the main methods persons with LIS communicate are with an analog alphabet board where the patient signals by blinking or making eye movement to an assistant [2]. This method is the most popular, though it is problematic because it gives the patient no privacy and he or she is dependent on an assistant. Other technological methods also exist; they are often, either a eye-tracking device or BCI a combined with a speech generator. The problem with today's technological methods is that they are very slow, so that the response from a LIS patient can have a time lag of 10-15 minutes when responding to anything from a joke to a comment [2]. For persons with CLIS, BCIs are their only way of communication, and unfortunately, there are no BCI communications devices that are ready to use for them [2].

BCIs have the potential to be an essential tool for persons not able to communicate by other means. As the study [2] tells, the social barriers from not being able to communicate can be more harmful than the physical condition. BCIs for communication may be the only option for persons with CLIS, though the technology is not fully developed or commercially available. In addition, current BCI applications for this target group are slow, have inadequate accuracy, and infringement of independence and privacy [2]. The difference in priority and opinion between end-users and researchers may also be why BCIs are not available for those who need them.

The study [22] surveyed the preferences for BCIs between persons with LIS (users), caregivers, and BCI researchers. A total of 85 persons participated in the questionnaire, each group equally represented, i.e., one-third were diagnosed with LIS, another third was caregivers, and the remaining were researchers. One of the questions was what they wanted a BCI application to be able to do, and here it was a general agreement between the three groups. *Direct personal communication, private conversation and writing* and *general computer use* was the top three ranked applications, *environmental control* was just behind the last of the top three. Another question was what kind of mental strategies they would prefer. Here it was more disagreement between the groups. The users and caregivers rated *attempted speech* highest, while the researchers rated *attempted hand movement (motor imagery)* highest.

As discussed in the study [22], the available applications may not be the most preferred, and what is favored and what is possible are not always aligned. A study [23] tries to help persons with LIS to have a more active role in their daily life by helping them communicate with a BCI. They have designed a two-way communication platform between a person with LIS and a caretaker. The person with LIS looks at a tablet, which shows different options, presented as colored icons, while recording their EEG signals. Classifying and deciding the options is done with eye movement and the color in focus. When an option is selected, the software notifies the caretaker. The caretaker can also give messages to the patient via the tablet. What is beneficial with this BCI is that it requires minimal user training, in contrast to implementing a motor imagery BCI [24]. Motor imagery is the imagination of movements without performing them. In addition, many people do not find motor imagery intuitive, it is challenging and difficult to learn [24]. Repeated acts of motor imagery can also make a person feel fatigued, as the act demands high cognitive effort [25]. A high fatigue level will also decrease the separability of features, which are essential for the classification in a ML method [26].

Creating a communication device for persons with CLIS is even more difficult, as they do not have any methods to communicate, so there is no feedback if the BCI works correctly. They no longer have any eye movement nor the option to gaze [27]. A study managed to create a platform for communicating with a CLIS patient [28] with a BCI. They established a binary communication based on mental imagery, which classified the EEG signals with Riemannian geometry. There is also research on BCIs to create actions in an environment, e.g., controlling a wheelchair [29]. The wheelchair controller was based only on eye movement, with six different states that were controllable. The eye movement was read with EEG and classified with Random Forest, a ML method.

3.2 EEG classification with CNNs

Various types of EEG classification methods were reviewed in the study [18], which also present an important note. Most of the reviewed methods were only evaluated in a closed offline environment. However, a BCI is usually only done in an online environment, real life. Implying, if the classification method does not work or cannot withstand real-life noise in an online environment, the BCI is useless. The review also discusses that the BCI pipeline often neglects the users' considerations. Taking these considerations into account, could ensure efficient BCI communication.

As mentioned in section 2.4, deep learning did not perform well when it comes to classification of EEG signals. However, the desire to get deep learning to work and make pre-processing redundant is still there. The hypothesis is that the pre-process loses valuable information which could help classification. A study [30] shows this precisely, by using a deep CNN, they managed to get good accuracies on a motor imagery dataset, by using raw EEG data as input. The study also shows a thorough discussion of different design choices for a CNN, the conclusion was that a deep CNN (several hidden layers) are needed. In addition, current state-of-the-art techniques for CNNs, such as batch normalization, which is normalizing the data in the training batch, and the activation function ELU, were highlighted as necessary. Another reason why deep learning on EEG signals should be researched: it can augment data and enlarge datasets by using Generative Adversarial Networks [31], which is needed, as EEG datasets usually are tiny.

Because of the limited EEG training data available, deep networks have problems with their performance. Shallow networks are more promising [18], as they have fewer parameters to train and therefore can perform well on small datasets. The study [20] created a single, more simple and shallow CNN structure, called EEGNet, see fig. 2.4. It classifies EEG signals that can work with different BCIs. They utilized depthwise and separable convolution layers, and the results became better than reference algorithms when limited data was available. The study also shows three different ways to visualize neurophysiologically features extracted from the model, which helps to understand the validity and robustness of the CNN model, compared to other DL datasets.

Since the other state-of-the-art machine learning techniques, such as ASVM and Riemannian geometry that have shown to be performing best on EEG classification [18], are out of the picture in this work. EEGNet is the neural network that has shown best results. By using different convolution methods instead of several hidden layers it manages to predict well with a smaller dataset. Depth in a neural network are usually what gives deep neural networks an advantage.

Chapter 4

Materials and methods

The steps to evaluate the design are visualized in fig. 4.1. The first step is recording EEG data with a protocol before pre-processing the data. Then, splitting the pre-processed EEG data to train a model evaluated with different metrics. This process either stops at the evaluation block or changes the hyperparameters with the help of Hyperband and trains a new model. The whole process was repeated for each subject in the dataset and for models based on a cross-subject dataset.

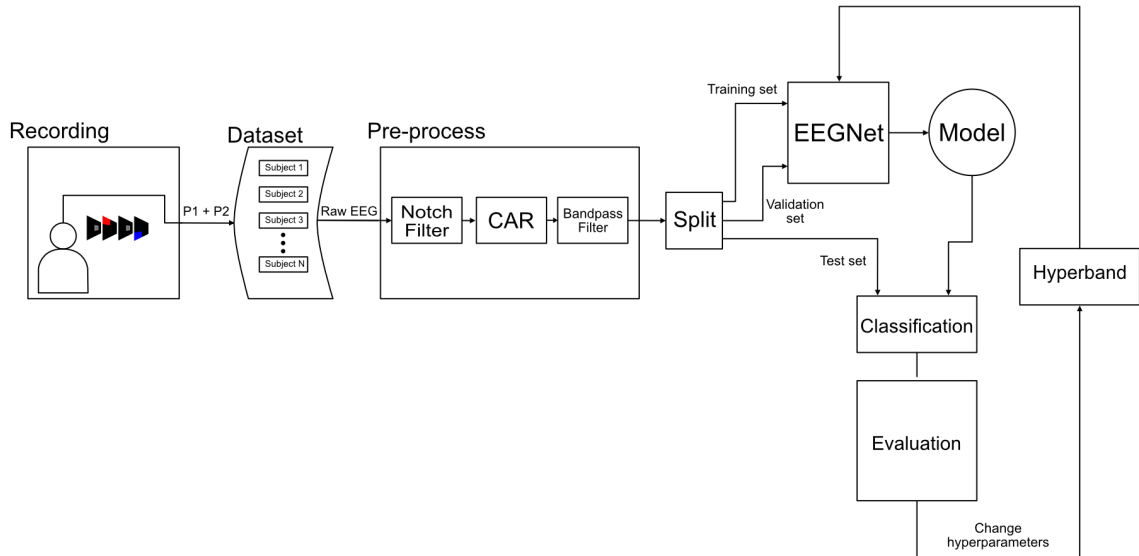


Figure 4.1: Flowchart of the process from data acquisition to model evaluation.

4.1 Hardware tools

OpenBCI headset

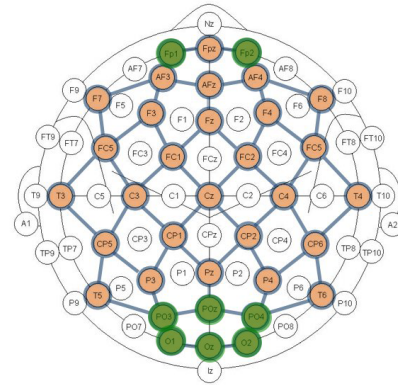
The EEG signals were recorded with an Ultracortex "Mark IV" EEG Headset, see fig. 4.2a, created by OpenBCI¹. The headset can record EEG data from up to 16 dry electrodes at 125 Hz. The recorded

¹<https://openbci.com>

data is sent to a computer via Bluetooth. If, however, the number of electrodes is kept at 8 or below, the sample rate is 250 Hz. The difference in sample rates come from the limitations of the Bluetooth bandwidth. The electrodes can be placed in 35 different locations, matching the international 10-20 system, see fig. 4.2b. The orange-colored circles are the available electrode placements on the headset in the international system. Two extra electrodes are being placed on the ear lobe. These serve as reference and ground, which aid in rejecting the common-mode noise from the environment.



(a) The Ultracortex "Mark IV" EEG headset



(b) Electrode placement on the headset, the used electrodes are marked with green.

Figure 4.2: The OpenBCI headset and electrode placement, colored orange and green, are viewed in the international 10-20 system. Both images are reprinted from [32].

The IDUN cluster

The high-performance cluster IDUN [33] was used to perform the data analysis. The cluster has more than 90 GPGPUs and 70 nodes, where each node contains two Intel Xeon cores and at least 128 GB of main memory. It is a platform for testing and prototyping high-performance computing software, graphic processing units simulations and design-space exploration. In this work the cluster was used to train models and find the best hyperparameters with a Hyperband search.

4.2 Electrode placement

The significant signals for this work are the brain's color perception and eye movement detection. From section 2.2 it is clear that the back part of the brain, the occipital lobe, and especially V4, is a crucial area to record EEG signals. V4 is hard to measure with non-invasive electrodes, but as an electrode records the sum of activation in its vicinity, the best method is to cover most of the area around it. Therefore, from the international 10-20 system, the electrode placements: *PO3*, *POz*, *PO4*, *O1*, *Oz*, and *O2* were chosen. Eye movement detection is recorded with electrodes placed close to orbita and the eye muscles, so therefore were *Fp1* and *Fp2* were chosen. See fig. 4.2b for a visualization of the node placement in the international 10-20 system, the used locations are colored green. Since it was possible to measure the two critical regions with only 8 electrodes, no more was used, as this would result in a sampling rate of 250 Hz. The reference and ground electrodes were placed on the right and left ear lobe, respectively.

4.3 EEG data collection

The desired BCI for this work is a program that shows several options on a computer screen at the same time, represented as colors or colors and icons, the separate tasks are visualized in fig. 4.3. The user of the BCI's eye movement and color focus decides what will happen. This design can be a modular communication platform for persons with LIS, as the options can be, e.g., letters used to write words, messages to the caretaker or environmental control. The study [23] inspires this design.

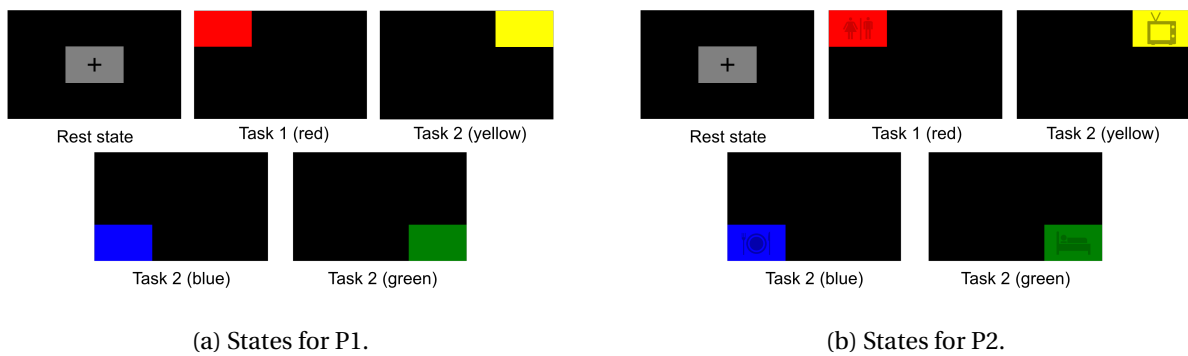


Figure 4.3: The different states for the two protocols.

4.3.1 Protocols

A general protocol, with five different states was designed to record the wanted EEG signals from volunteers (see fig. 4.3). A gray box in the middle representing the resting state. A cross was added to this box to help the participants concentrate on the rest state. The four other states are classified as task-states and are colored boxes in each corner on the screen. The upper left is colored red, the upper right is yellow, the lower left is blue, and the lower right is green. The tasks state appeared one at a time in a random sequence, and following a task state was always the resting state. The rest and task state were visible for 2 seconds before switching to a new state, depending on the prior state, visualized in fig. 4.4. The subjects moved only their eyes and focused on the state presented on the screen. There was an equal amount of the different tasks in each session.

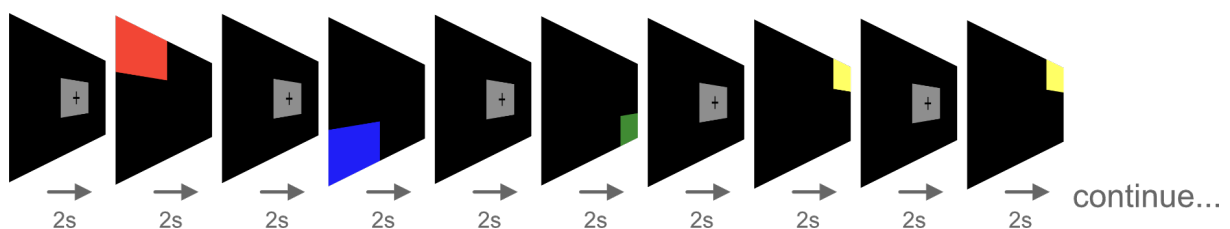


Figure 4.4: A sample of how the protocol would run. Here only 5 tasks are shown, and the states are from P1. P2 looks the same, only is using its own states with an icon on each colored box, shown in fig. 4.3b.

There were two main reasons behind the choice of 2 seconds viewing time. The first is that the input to a CNN has to be equal. A significant fraction of the data points would have been useless if the periods had varied. The second reason was that a shorter period would decrease the possibilities

to split the samples, which could be a method to enlarge the dataset. While also, 2 seconds would include enough data points to train on, and a more extended period is not preferred if the BCI shall be used to communicate. Since the continuous implementation would have to wait over 2 seconds before a state could be classified.

The general protocol explained in the previous paragraph was used as a base to create two different protocols. From now on, referred to as protocol 1 (P1) and protocol 2 (P2). How a task are visualized are the difference between them. P1 (fig. 4.3a) only showed colored boxes, while P2 (fig. 4.3b) showed the same colors in addition to an icon. The four icons, visualized in the figure, resembled the need to use the bathroom (task 1), want to be entertained (task 2), hungry (task 3), and tired (task 4).

All volunteers were informed to sit relaxed during the recordings and try to move as little as possible. They were instructed only to move their eyes, nothing else. If they needed to blink, they were instructed to do it when the resting state was showing, as there are a lot more data on the resting states than the tasks states. Each subject had the choice of completing 4 or 6 sessions. Each session alternated between P1 and P2 in a sequence, starting with P2, then P1. One session took 320 seconds, which includes 20 recordings of each task state, a total of 80 task states, and 79 rest recordings since the first sample was removed because it was too noisy.

4.3.2 Metadata of the dataset

A total of 33 healthy participants brain signals were recorded while they perform the tasks in the protocols. There was 54.5% male participant, and the average age was 24.9 years old, see fig. 4.5. Most of the participants chose to do 6 sessions; only 9 chose 4 sessions, totaling 180 sessions. One of the participants, number 3, was color blind and had trouble differentiating between red and green.

There are, however, problems related to the dataset and the data acquisition with the OpenBCI headset. There is a brief period between the headset starts recording to the first markers are placed, a problem of real-time computing. The software can also continue to record after the protocol has finished running, though this is not an issue, as these data points are not being used. The final problem is that the headset might lose or corrupt some bytes during the recordings when sent via Bluetooth, resulting in different amounts of data points at each session. The difference is showcased in fig. 4.6, where a wider opaque blue implies more sessions with that number of data points. In addition, there are several seconds of difference between the protocols. Even though they were run on the same length, all sample rates were equal, and all session markers were the same.

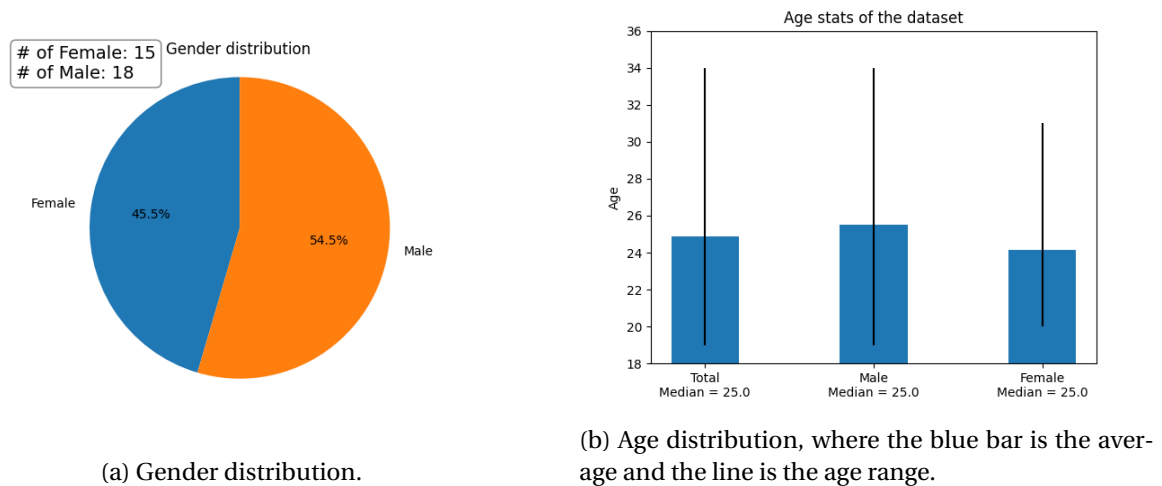


Figure 4.5: The general distribution of the dataset.

Another problem is that the OpenBCI headset does not always record at 250 Hz. The problem was solved by the thread that was connected to the headset kept track of the time it used to collect the EEG signals. Resulting in removing three sessions as they had a different sampling rate than 250 Hz. The rest was kept in the complete dataset.

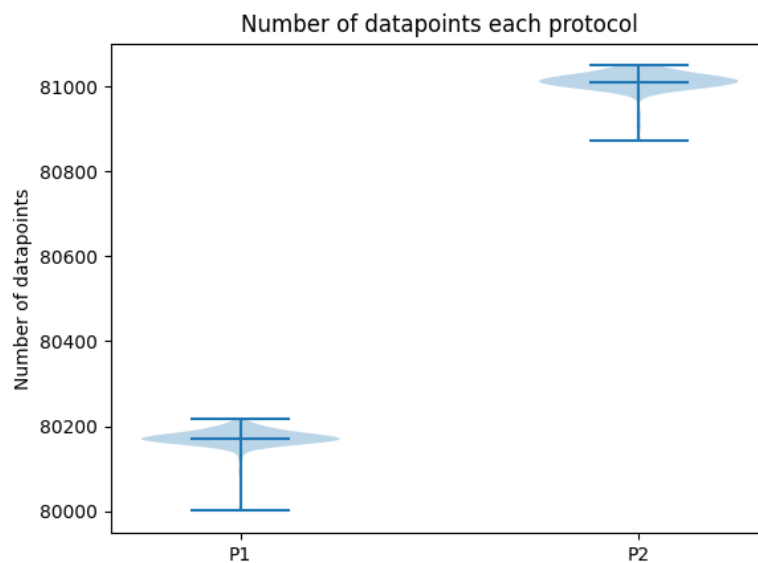


Figure 4.6: Sessions from each protocol versus how many data points got recorded.

4.4 Pre-Processing

The raw signals can be very noisy, and as discussed in section 2.3.3, it is easy to fill the signal with artifacts. Therefore it is necessary to enhance the signal-to-noise ratio before analyzing the EEG signals.

Notch filter

The EEG signal can be influenced by noise from the environment, such as a common-mode signal. This signal is noise around 50 Hz, here in Europe. Removing this noise can be done with a band-stop filter, which stops a range of frequencies from passing based on two cut-off frequencies. In this work, a filter is a mathematical procedure that lets some frequencies pass, decided by its cut-off frequencies. Here it is only needed to stop a narrow range of frequencies, or rather just one frequency. So removing the common-mode signal can be done with a Notch filter, a band-stop filter, just with a narrow stopband.

Common Average Reference

The reference node should be independent of the other electrodes to reduce the noise, as mentioned in section 2.3. As this is not possible, creating a synthetic reference can help with enhancing the signal-to-noise ratio, where Common Average Reference (CAR) [14] is such a method. CAR removes simultaneously-recorded common information from all electrodes. It removes that noise from each channel, V_i^{CAR} , where i is the number of a channel, it is computed with the formula:

$$V_i^{CAR} = V_i^{ER} - \frac{1}{n} \sum_{j=1}^n V_j^{ER} \quad (4.1)$$

Where V^{ER} is the potential between the i^{th} electrode and the reference, and n is the number of electrodes.

Bandpass filter

A way to reduce the noise is to filter the signal, e.g., a bandpass filter. A bandpass filter has two cut-off frequencies, an upper and a lower limit. Only the frequencies between the limits are allowed to pass through. The EEG signals used in this work are bandpass filtered with a lower limit of 0.5 Hz and an upper of 50 Hz.

Dataset splitting and balancing

The dataset was split into three parts: a training set, a validation set, and a test set to ensure an unbiased evaluation. The training set got 70% of all the samples, while the validation and test set have 15% each. The splits are done pseudorandomly, the splitting is random, but the splits is the same each time it is being done. The pseudorandom splits ensured that the samples used in training and evaluation did not affect the performance of the classifiers. As there are not that many samples in the dataset, each split got balanced, so it contained an equal amount of each class. An effort was made to enlarge the current dataset, so the 2-seconds sample containing a specific state was divided by two. So, effectively multiplying the number of samples by a factor of two, though each sample now contains less information.

4.5 EEGNet architecture

In this work, the CNN EEGNet [20], is used to classify the different states in the EEG signal. The strength of EEGNet is its usability on different BCIs, performance on limited data, and its ability to produce neurophysiologically interpretable features. Its complete architecture can be viewed in table 4.1 and as a simplified version in fig. 2.4. The main thing to notice in its architecture is that it uses depthwise and separable convolution. According to the authors in [20], a depthwise layer will help learn spatial filters for each temporal filter, enabling efficient extraction of frequency-specific spatial filters. A spatial filter refers to a space-dependent filter, while a temporal filter refers to a time-dependent filter. The hyperparameters F_1 and D decide how many temporal and spatial filters the network shall train, respectively.

Block	Layer	# filters	Size	# params	Output	Activation	Options
1	Input				(C, T)		
	Reshape				(1, C, T)		
	Conv2d	F_1	(1, 64)	$64 \cdot F_1$	(F_1, C, T)	Linear	Mode = Same
	BatchNorm			$2 \cdot F_1$	(F_1, C, T)		
	DepthwiseConv2D	$D \cdot F_1$	(C, 1)	$C \cdot D \cdot F_1$	$(D \cdot F_1, 1, T)$	Linear	Mode = valid, depth = D, Max norm = 1
	BatchNorm			$2 \cdot D \cdot F_1$	$(D \cdot F_1, 1, T)$		
	Activation				$(D \cdot F_1, 1, T)$	ELU	
	AveragePool2D		(1, 4)		$(D \cdot F_1, 1, T // 4)$		
	Dropout				$(D \cdot F_1, 1, T // 4)$		p = 0.25 or p = 0.5
2	SeparableConv2D	F_2	(1, 16)	$16 \cdot D \cdot F_1 + F_2 \cdot (D \cdot F_1)$	$(F_2, 1, T // 4)$	Linear	Mode = Same
	BatchNorm			$2 \cdot F_2$	$(F_2, 1, T // 4)$		
	Activation				$(F_2, 1, T // 4)$	ELU	
	AveragePool2D		(1, 8)		$(F_2, 1, T // 32)$		
	Dropout				$(F_2, 1, T // 32)$		p = 0.25 or p = 0.5
	Flatten				$(F_2 \cdot (T // 32))$		
Classifier	Dense	$N \cdot (F_2 \cdot T // 32)$			N	Softmax	Max norm = 0.25

Table 4.1: The EEGNet architecture copied from [20], where C = number of channels, T = number of time points, F_1 = number of temporal filters, D = depth multiplier (number of spatial filters), F_2 = number of pointwise filters, and N = number of classes.

The separable convolution helps with separating the learning, how to summarize individual feature maps, with how to optimally combine the feature maps. In the separable convolution, it is the pointwise filter, set by the hyperparameter F_2 , that finds the optimal combinations of the feature maps. If F_2 is equal to $F_1 \cdot D$, it learns an equal amount of feature maps as its input. If F_2 is lower, it learns fewer, implying a compressed representation, while a higher number implies an overcomplete representation.

The study showed that this network managed to get good accuracies without pre-processing the raw EEG signals. In this work, however, the input to the network will be a pre-processed EEG signal to ease the training. The loss functions used were *binary cross-entropy* or *categorical cross-entropy*, de-

pending on the number of classes, while the network was optimized with the *Adam* algorithm.

4.6 Methods for evaluation

Evaluation of the model is being done in two different steps: while a model is training and after the same model has finished training. While a model is training, the evaluation is computed with the validation set after each epoch. This evaluation validates how the model is doing while it is training and indicates if it is starting to overfit or not. When the training is finished, the model predicts on a set of samples the model has never seen before, i.e., the test set. This prediction gives an unbiased evaluation of the model.

The predicted labels are compared with the actual labels and create a confusion matrix (fig. 4.7) which visualize the performance of the model. There are two different layouts of the confusion matrix: for binary classification (fig. 4.7a) and multiclass classification (fig. 4.7b), which shows what the different regions will be for C_1 . Its important to note that the diagonal is always the number of true predictions for the corresponding class, while the rest of the row and column at that class index is False Negative and False Positive, respectively, the rest of the matrix is True Negative.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (<i>TP</i>)	False Negative (<i>FN</i>)
Actual Negative	False Positive (<i>FP</i>)	True Negative (<i>TN</i>)

(a) Confusion matrix for a binary classifier

		Predicted labels			
		C_1	C_2	C_3	C_n
True labels	C_1	TP		FN	
	C_2			...	
	C_3	FP		TN	
	C_n			...	

(b) Confusion matrix for a multiclass classifier

Figure 4.7: Confusion matrices for a classifier. Note that fig. 4.7b show the regions for class 1, C_1 .

From the confusion matrix, several evaluation metrics may be computed. One is the accuracy (eq. (4.2)), which calculates the fraction of how many samples were correctly calculated. Another is Recall, also called sensitivity, calculates the fraction of how many positive samples were predicted correctly. Specificity (eq. (4.3)) is similar to recall, only that it looks at the negative samples instead of the positive. The final metric used in this work are the F1-score (eq. (4.4)), which is the harmonic mean between recall and precision, where precision is the fraction of correct positive predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.2)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4.3)$$

$$F1 = \frac{2 * TP}{2TP + FP + FN} \quad (4.4)$$

Chapter 5

Experimental results

The experiments are performed with the dataset created by the protocols. Their evaluation will decide if the chosen EEG signals are usable, i.e., the task states are differentiable, and if the BCI design can be implemented. The evaluation is done by creating models with EEGNet and evaluating them with the metrics accuracy, F1, and specificity. Two different classification models are created, one that differentiates a sample between "task" and "rest", and one that classifies the four different tasks. These two classifications will be referred to as **resting-state vs task** and **4-class**. Two separate classification models simplify the classification and will positively affect real-time implementation. The computational cost will be reduced in real-time since the two-class model is mainly classifying, and only if it detects a task, a larger model starts computing. In addition, removing one class (5-class to 4-class) increases a model's performance.

The two classification models will be evaluated with different evaluation metrics. The **resting-state vs task** models will be evaluated with the three evaluation metrics, whereas **4-class** with accuracy and F1, since specificity indicates if a model predicts "task", when it should be "rest", which is not desired. This evaluation is not necessary in a **4-class** classification as it is more critical to predict correctly and measure with accuracy.

The entire dataset splits into three subsets based on the different protocols, where the subset includes data from just P1 or P2, or from both. In addition, the data from the sessions are mixed before splitting into three new sets: training, validation, and test set, as explained in section 4.6. The hyperparameters used were the same as the standard ones in EEGNet's study [20]. Each model had a max epoch number of 500, but since early stopping was activated, the training length varied on each model. If nothing else is stated, the hyperparameters to the model are the same as used in EEGNet's study.

5.1 Evaluation of subject performance

The object of the first experiment was to see how good the individual data from the different subjects were, in addition, to examine if some subject data did better than others, if so, these subjects could be selected when creating cross-subject models. The length of each input sample is 2-seconds, i.e., the whole period a state was viewed by a subject. The graphs presented in this subsection are read in

the following way. The red dashed line on top of the bars represents the chance level, metrics below that line have performed worse than chance. In the **resting-state vs task** plots, the specificity should not be very different from the two other metrics, and a high specificity are preferred over a low one.

5.1.1 Individual models based on P1 data

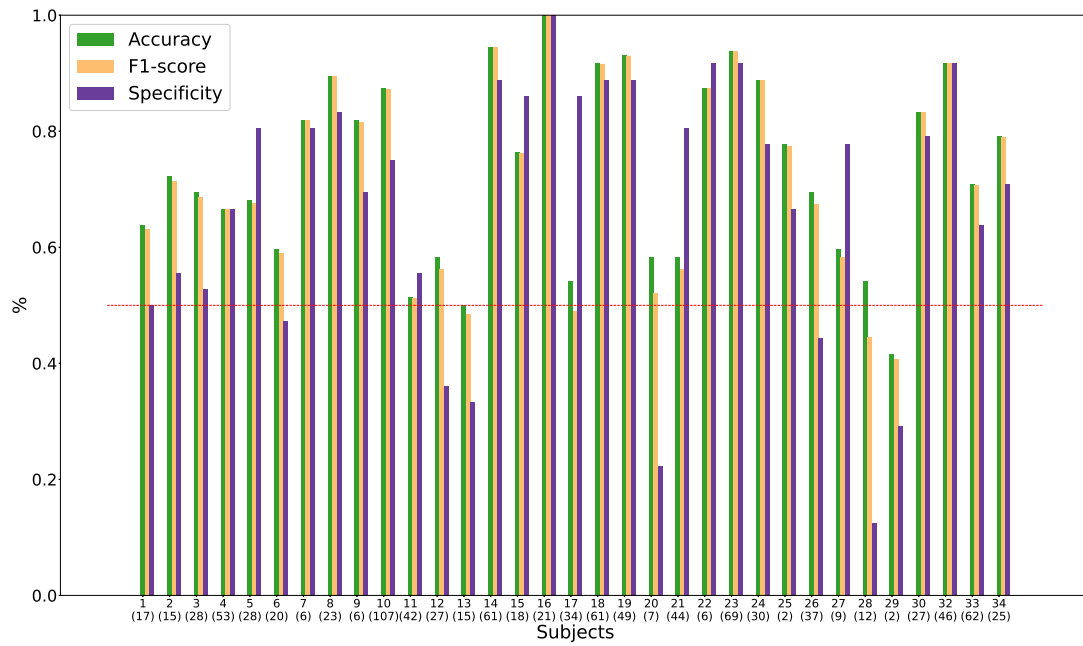
Here, the classifier for each subject are created with data from P1. The results from this can be seen in fig. 5.1a and fig. 5.1b, which shows **resting-state vs task** and **4-class** classification, respectively. The subfigures in fig. 5.1 show that almost all subjects got a model that managed to predict the classes. The subject models performing best on both classifiers are 7, 8, 18, 32, and 34. In addition are the models for the subjects 2, 3, 4, 5, 9, 14, 15, 16, 19, 22, 23, 24, 25, 27, 30, and 33 performing relative well on both classifiers. The model to subject 16 managed to predict perfectly on **resting-state vs task**, while subject 25 had the best model for **4-class** prediction.

5.1.2 Individual models based on P2 data

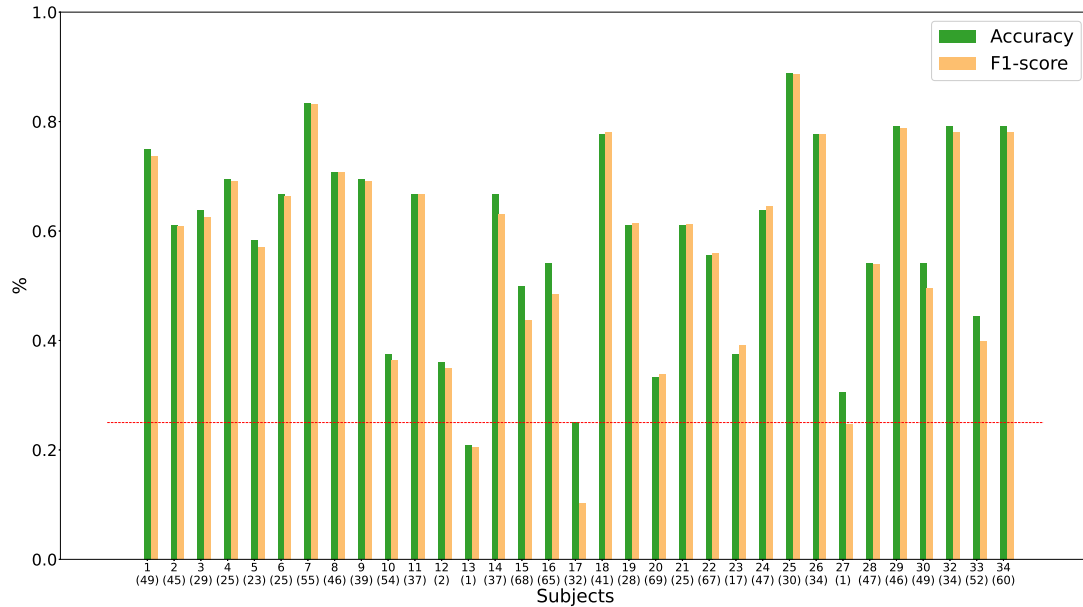
In this experiment the individual classifiers were created with data from P2. From the results in fig. 5.2, it is immediately apparent that the models had trouble classifying this data. Only a few models manage to predict better than the chance level on both classification problems, namely the subjects 6, 15, 25 and, 33. Note also the lack of coherence between each subject model pair. For example, subject 21 performed best on its **4-class** model, while its **resting-state vs task** model performed worse than chance. The **resting-state vs task** model for subject 8 has a very high specificity, while the other two metrics are low, meaning that the prediction mostly was only "rest".

5.1.3 Models based on P1 and P2 data

Finally, individual models were trained on data from both P1 and P2, results presented in fig. 5.3. First thing to note is that more models in fig. 5.3b than in fig. 5.3a performs better than chance. Only five classifier pairs performed relatively well on both classifiers, namely subjects 2, 11, 22, and 34. It is also possible to argue that the subject models 15, 29, and 30 have a relatively good performance. Their only problem is low specificity, which was generally the case for the other results as well.

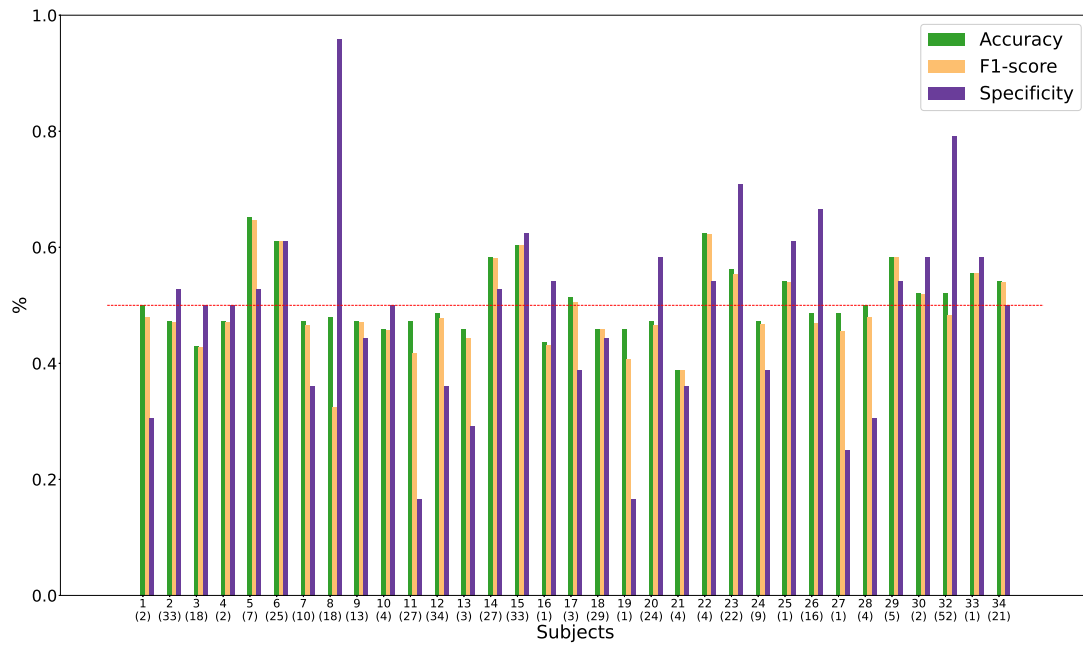


(a) Individual **resting-state vs task** classification with P1 data and 2 seconds samples.

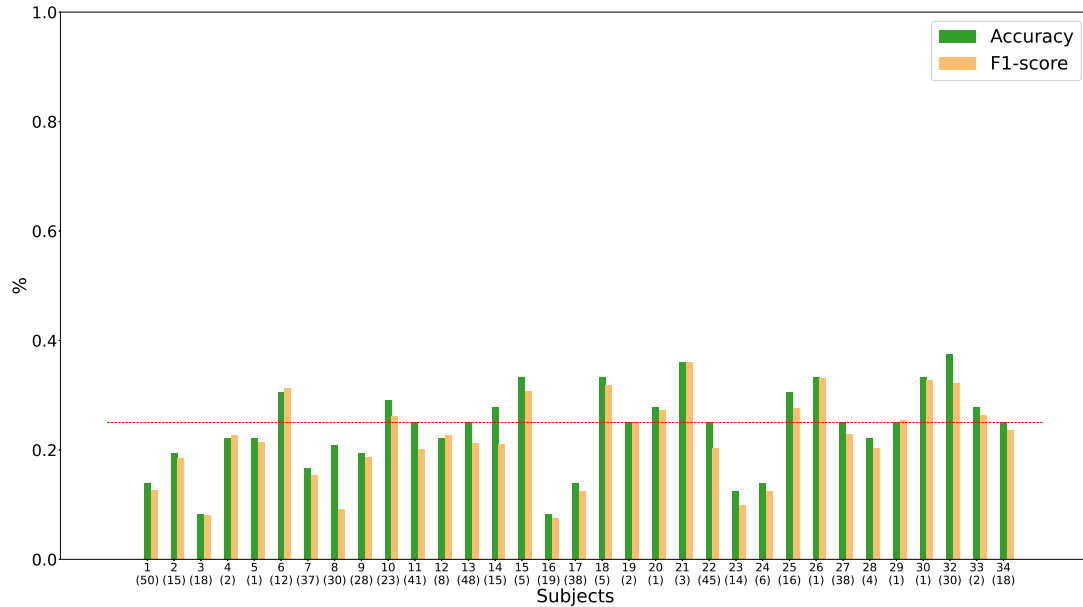


(b) Individual **4-class** classification with P1 data and 2 seconds samples.

Figure 5.1: Individual **resting-state vs task** and **4-class** models were created for each subject with data from P1. Each sample is 2 seconds long, 500 data points. The red line illustrates the level of chance. The number in brackets below the subject number, identify the number of epochs the model was trained. The y-axis goes from 0 to 1, which responds respectively to 0% and 100%.

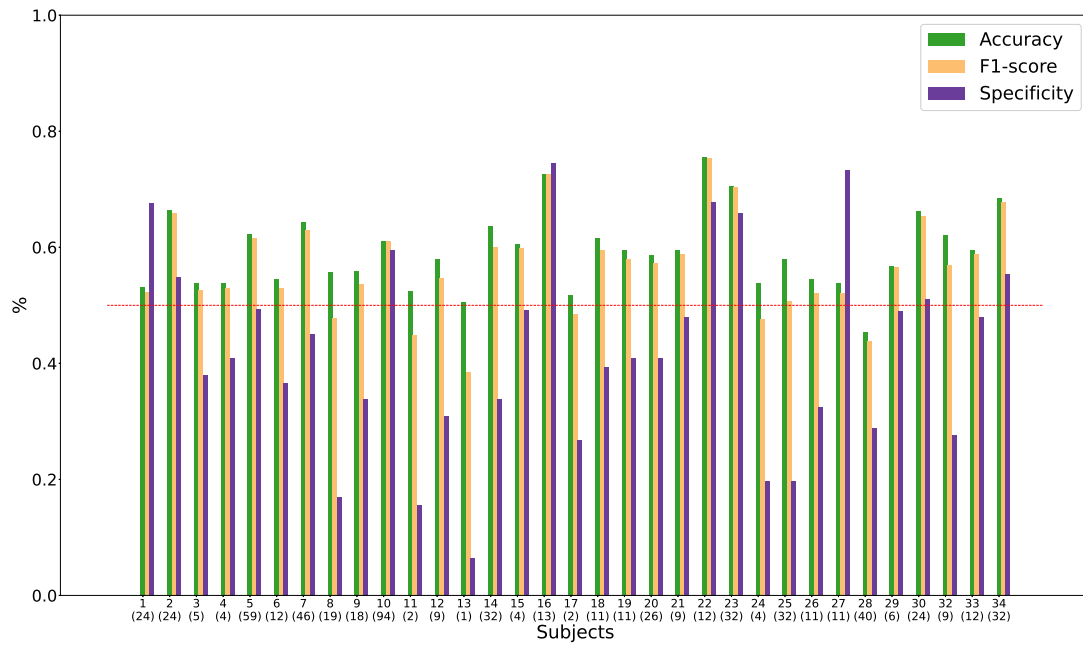


(a) Individual **resting-state vs task** classification with P2 data and 2 seconds samples.

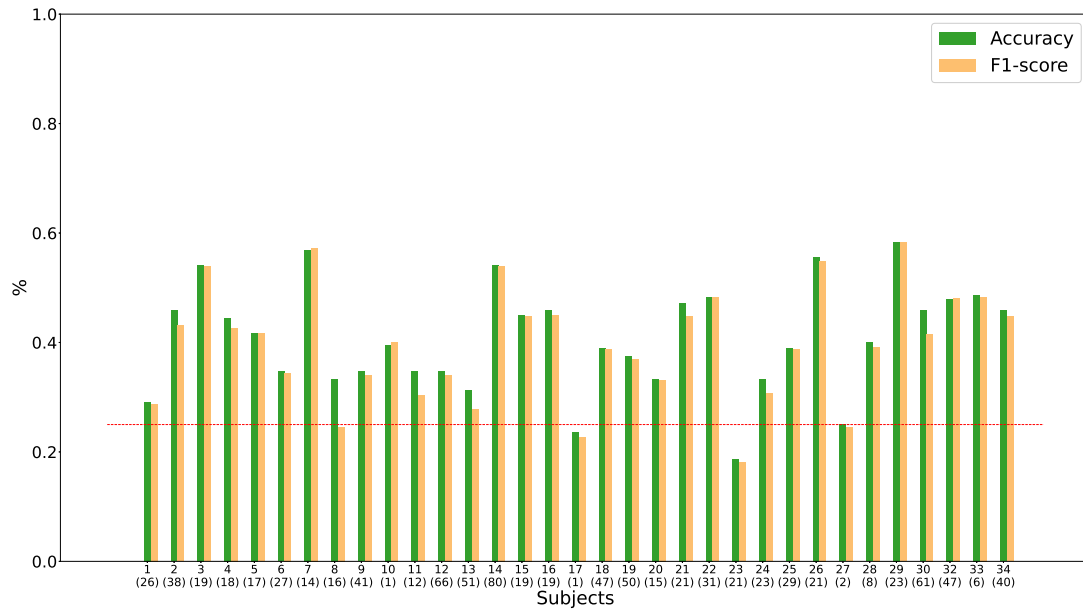


(b) Individual **4-class** classification with P2 data and 2 seconds samples.

Figure 5.2: Individual **resting-state vs task** and **4-class** models were created for each subject with data from P2. Each sample is 2 seconds long, 500 data points. The red line illustrates the level of chance. The number in brackets below the subject number, identify the number of epochs the model was trained. The y-axis goes from 0 to 1, which responds respectively to 0% and 100%.



(a) Individual **resting-state vs task** classification with P1 and P2 data and 2 seconds samples.



(b) Individual **4-class** classification with P1 and P2 data and 2 seconds samples.

Figure 5.3: Individual **resting-state vs task** and **4-class** models were created for each subject with data from both P1 and P2. Each sample is 2 seconds long, 500 data points. The red line illustrates the level of chance. The number in brackets below the subject number, identify the number of epochs the model was trained. The y-axis goes from 0 to 1, which responds respectively to 0% and 100%.

5.2 Evaluation of subject performance with a sample length of 1 seconds

The experiments in this section are very similar to the ones in section 5.1. The objective is to see if it is possible to reduce the sample length and still get good accuracy, as this would increase the speed of the BCI. So the sample length is 1-second instead of 2-seconds, which effectively doubles the dataset with samples. However, each sample will contain less information, and only half of the samples will include eye movement. Models are created on all subjects to see if there is a difference from the results in section 5.1 with the individual subject data.

5.2.1 Models based on P1 data

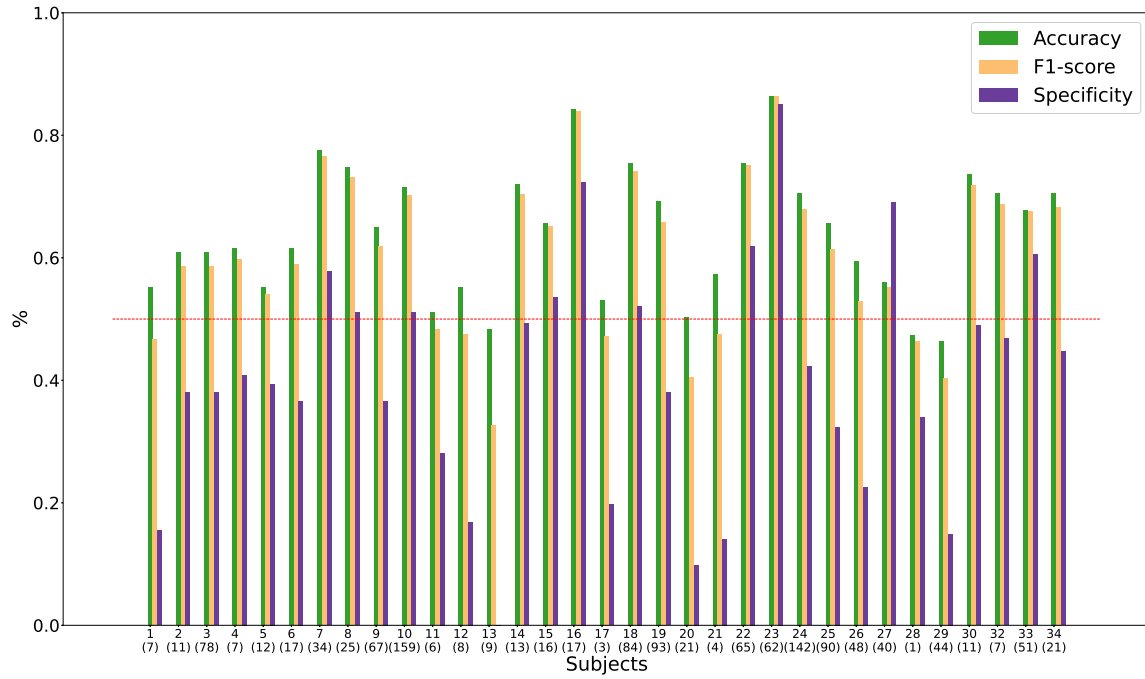
The results of the individual models created with P1 data can be viewed in fig. 5.4. The first thing to note is the generally low specificity on the **resting-state vs task** plot, while nearly all the **4-class** models have a score higher than the chance level. The classifier pairs that performed better than chance on both classifiers are 7, 15, 18, 22, 23, and 34.

5.2.2 Models based on P2 data

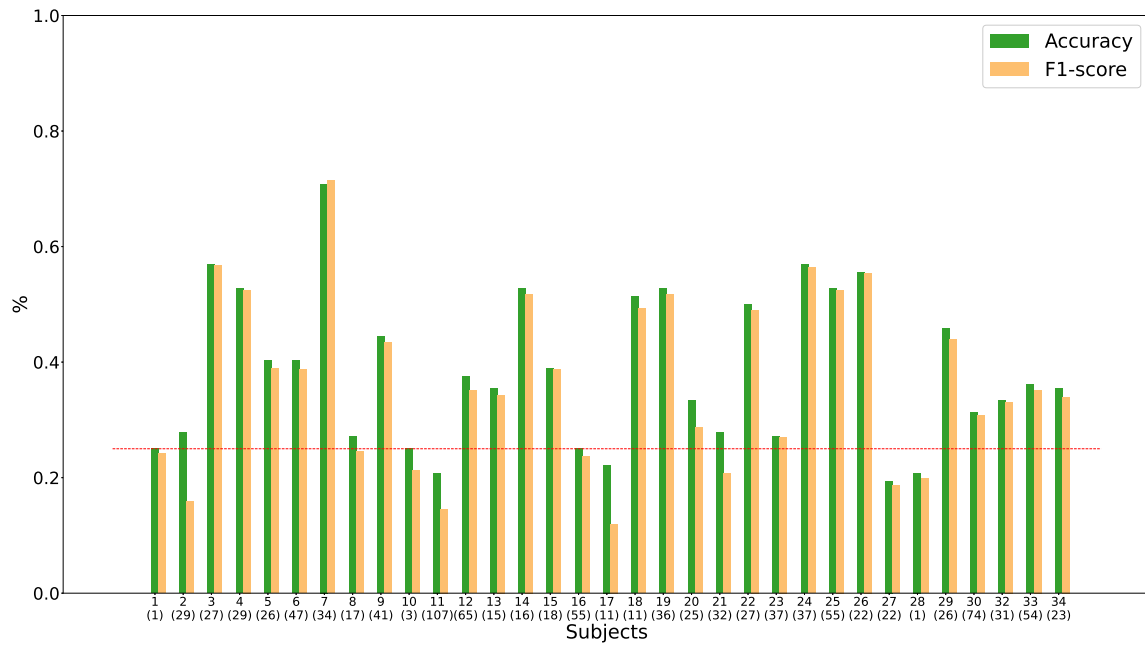
The results for the individual models is presented in fig. 5.5, and it is evident that only a few classifier pairs managed to predict better than the chance level. Only one pair managed to predict better than chance on both, namely subject 15.

5.2.3 Models based on P1 and P2 data

The results for the models created with both P1 and P2 are presented in fig. 5.6. In fig. 5.6a are the specificities either very low or very high, implying that the models do not manage to differentiate the two classes. The only classifier pairs performing relatively well on both classifiers are subject number 4, 22, and 29.

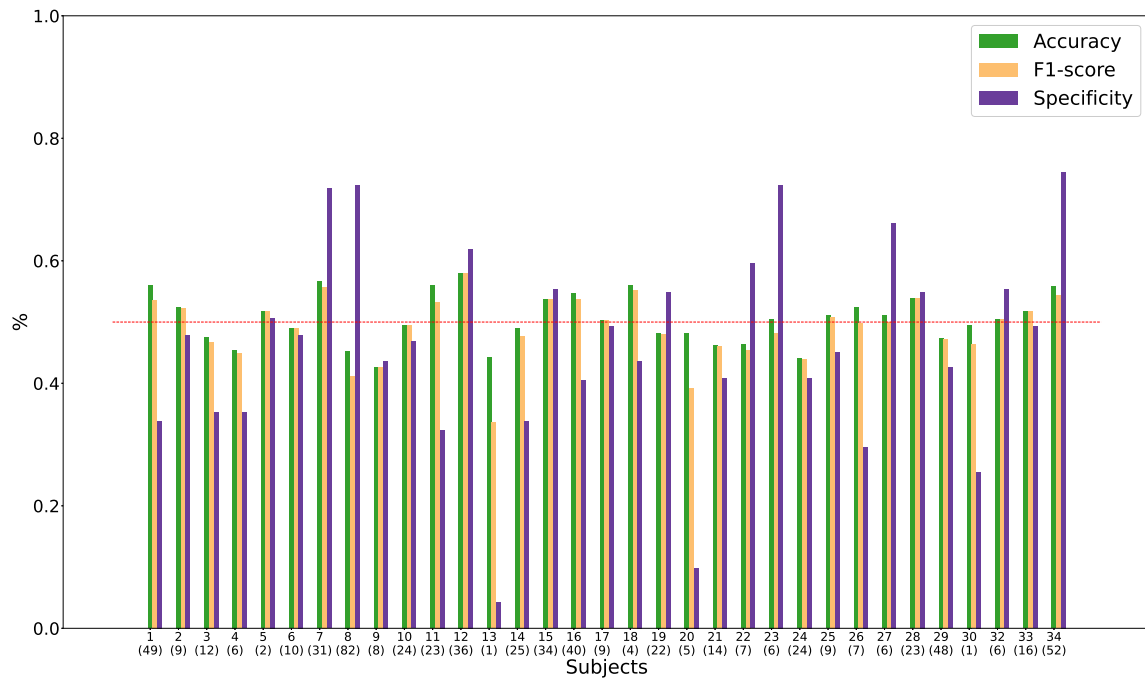


(a) Individual **resting-state vs task** classification with P1 data and samples with a period of 1 second.

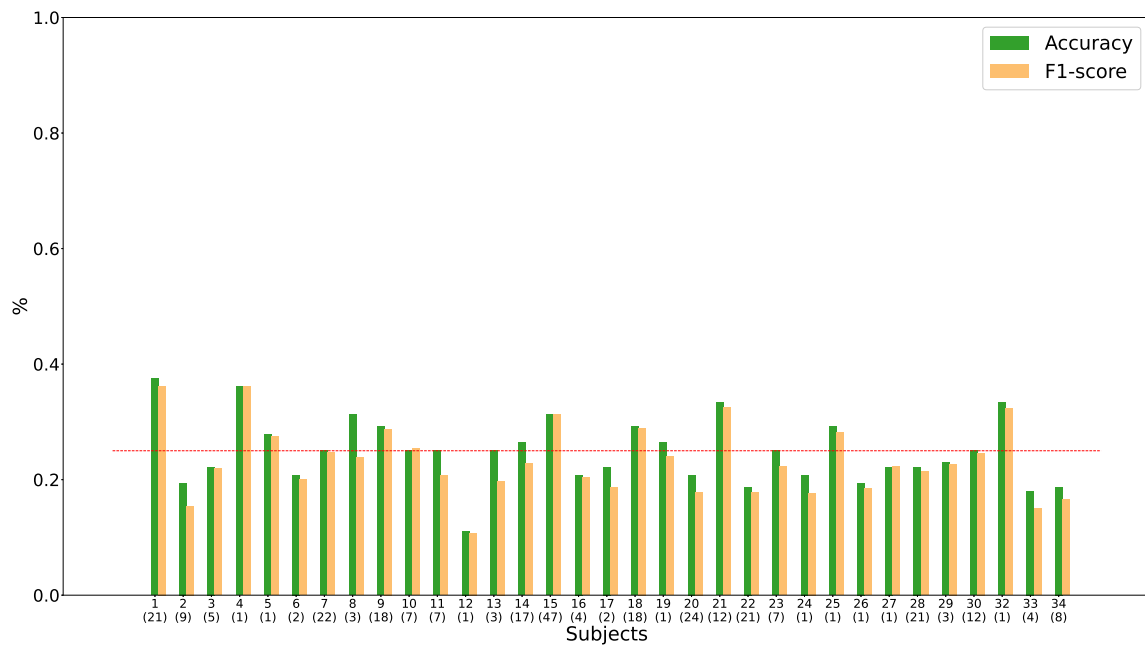


(b) Individual **4-class** classification with P1 data and samples with a period of 1 second.

Figure 5.4: Individual **resting-state vs task** and **4-class** models were created for each subject with data from P1. Each sample is 1 seconds long, 250 data points. The red line illustrates the level of chance. The number in brackets below the subject number, identify the number of epochs the model was trained. The y-axis goes from 0 to 1, which responds respectively to 0% and 100%.

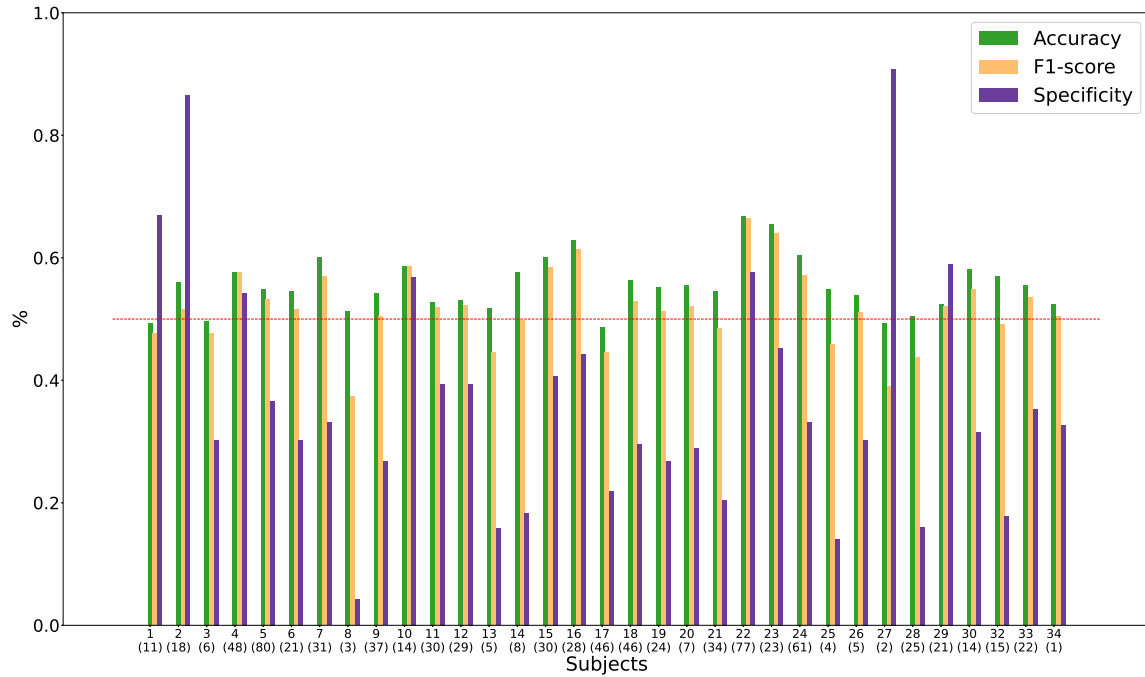


(a) Individual **resting-state vs task** classification with P2 data and samples with a period of 1 second.

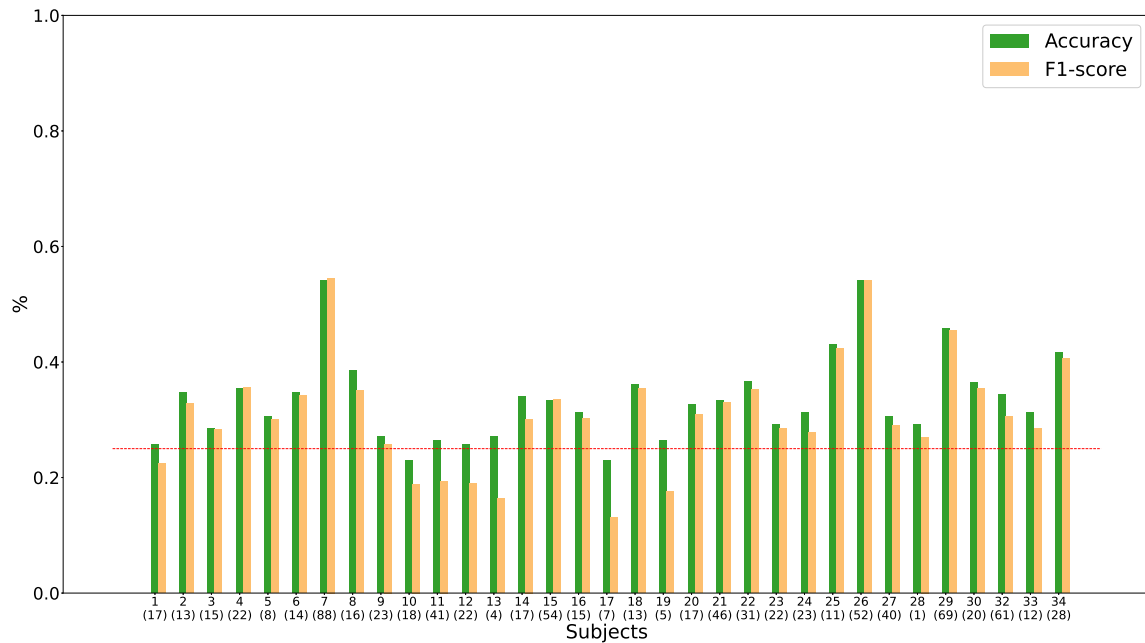


(b) Individual **4-class** classification with P2 data and samples with a period of 1 second.

Figure 5.5: Individual **resting-state vs task** and **4-class** models were created for each subject with data from P2. Each sample is 1 seconds long, 250 data points. The red line illustrates the level of chance. The number in brackets below the subject number, identify the number of epochs the model was trained. The y-axis goes from 0 to 1, which responds respectively to 0% and 100%.



(a) Individual **resting-state vs task** classification with P1 and P2 data and samples with a period of 1 second.



(b) Individual **4-class** classification with P1 and P2 data and samples with a period of 1 second.

Figure 5.6: Individual **resting-state vs task** and **4-class** models were created for each subject with data from both P1 and P2. Each sample is 1 seconds long, 250 data points. The red line illustrates the level of chance. The number in brackets below the subject number, identify the number of epochs the model was trained. The y-axis goes from 0 to 1, which responds respectively to 0% and 100%.

5.3 Evaluating individual P2 data on individual subject P1 models

The results in section 5.1 and section 5.2 show that the individual models created with data from P1 performed a lot better than the models created with P2. The advantage of using the P2 states are that they show what the state means, which can be beneficial, as the user of the BCI would not need to remember what the states do. So, this experiment will try to improve the individual results on the P2 dataset by using the trained models shown in fig. 5.1, and the same test data used to evaluate the results in fig. 5.2. This experiment is limited by only using the subjects with the best performance on both classifiers created with P1 data, namely the subjects: 7, 8, 18, 32, and 34. The results from this experiment are presented in fig. 5.7. Only the **4-class** classifier, fig. 5.7b, got on average better results than its counterpart, while there was a really low difference for the **resting-state vs task** classifier, fig. 5.7a. However, the difference was generally insignificant on both classifiers, except the **4-class** model for subject 7.

Subject	Accuracy		F1-Score		Specificity	
	New	Diff.	New	Diff.	New	Diff.
7	0.444	-0.028	0.444	-0.022	0.444	0.083
8	0.562	0.083	0.558	0.234	0.458	-0.500
18	0.458	0	0.434	-0.024	0.25	-0.194
32	0.479	-0.042	0.423	-0.060	0.167	-0.625
34	0.5	-0.042	0.497	-0.044	0.417	-0.083

(a) **Resting-state vs task** prediction on individual P2 data with individual P1 models.

Subject	Accuracy		F1-Score	
	New [%]	Diff.	New [%]	Diff.
7	0.583	0.416	0.586	0.433
8	0.292	0.084	0.272	0.179
18	0.361	0.028	0.358	0.039
32	0.5	0.125	0.5	0.178
34	0.167	-0.083	0.167	-0.069

(b) **4-class** prediction on individual P2 data with individual P1 models.

Figure 5.7: Results from predicting classes from the individual P2 data with a P1 model. The difference (Diff.) are calculated by subtracting the new ones with the subjects score in fig. 5.2. The scores that are better when using the P1 model are highlighted green.

5.4 Cross-subject evaluation

The goal of this experiment was to observe if a generalized model could be constructed by using data from all subjects. With a general model based on several subjects, it will be easier to add more subjects to the model, and it may be possible to use the model without further train the model with data from the new subject. Four different models were trained on each data subset and classification problem. The models were the default EEGNet with 2-seconds samples, default EEGNET with 1-second samples, Hyperband tuned EEGNet with 2-seconds samples, and, lastly, Hyperband tuned EEGNet with 1-second samples. Due to time restrictions, the Hyperband algorithm had only 48 hours to find the best hyperparameters, meaning that it did not finish its search.

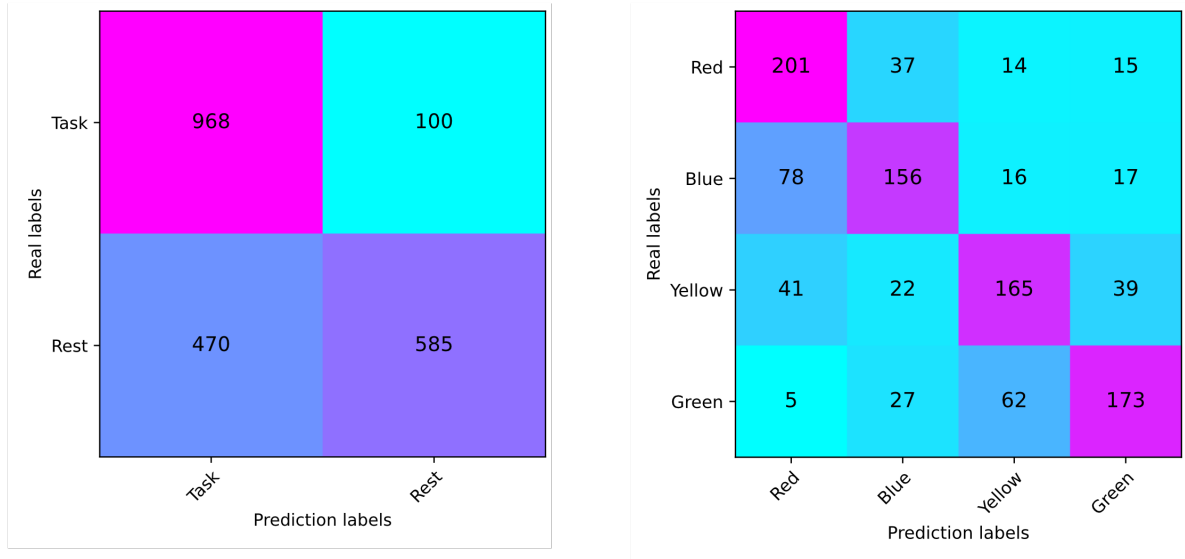
5.4.1 Models trained with P1 data

The result from the different models trained with P1 data is presented in table 5.1. The first thing to note is that for **resting-state vs task**, the standard EEGNet outperformed the model found with Hyperband, though this is not the case for the **4-class** models. The model that performed best for **resting-state vs task** where the standard EEGNet model with 2-seconds, though the specificity was not very high. The low specificity is also apparent in its confusion matrix, fig. 5.8a, which shows that the "task" state has been predicted more frequent than the "rest" state. The best **4-class** model is the

one tuned with Hyperband, and trained on sample with a length of 2-seconds. In its confusion matrix, fig. 5.8b, is it possible to see that the model was most uncertain with the "red" and "blue" classes, in addition to having trouble with differentiating between "red" and "blue", and "yellow" and "green".

Model	Rest Vs Task			4 Class	
	Accuracy	F1-Score	Specificity	Accuracy	F1-Score
EEGNet 2s	0.732	0.723	0.555	0.491	0.458
EEGNet 1s	0.639	0.612	0.378	0.415	0.394
Hyp. EEGNet 2s	0.716	0.705	0.52	0.651	0.650
Hyp. EEGNet 1s	0.579	0.521	0.232	0.312	0.247

Table 5.1: The scores for cross subject models, created with data from P1. The score that are highlighted yellow, are the best for that metric. Here Hyp. means that the hyperparameters are found with Hyperband, and 2s and 1s means a sample length are 2-seconds and 1-second, respectively.



(a) **Resting-state vs task** confusion matrix for the model EEGNet 2s created with P1 data.

(b) **4-class** confusion matrix for the model Hyp. EEGNet 2s created with P1 data.

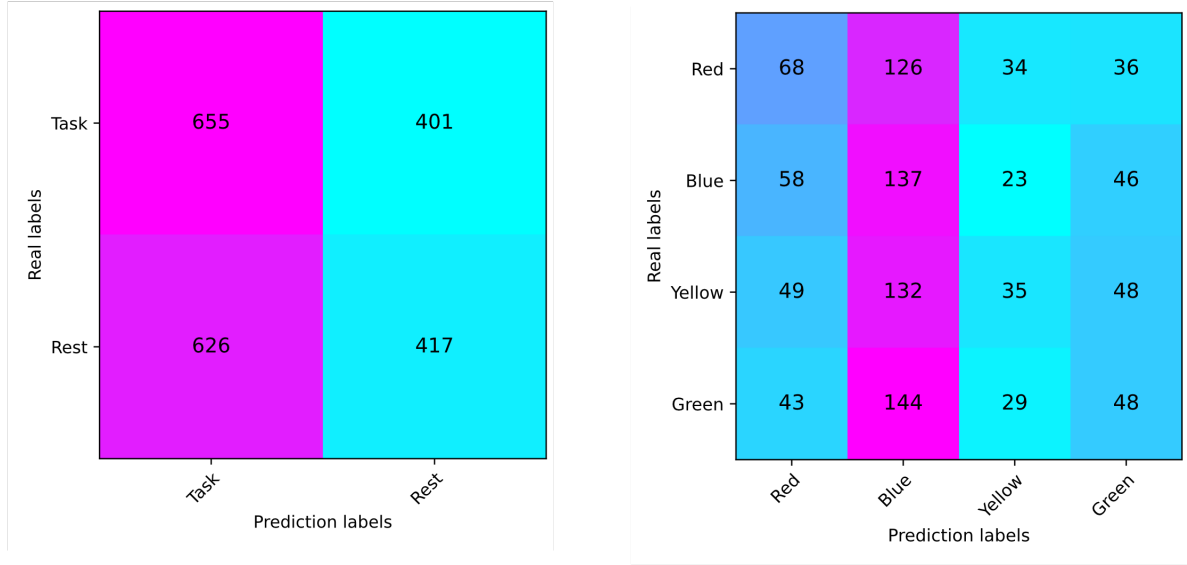
Figure 5.8: Confusion matrices for the best cross subject model for the two classification problems, when the models are trained with P1 data.

5.4.2 Models trained with P2 data

The result from the different models trained with P2 data is presented in table 5.2. Only one model managed to have accuracy above the chance level, namely, the model tuned with Hyperband and with a sample length of 2-seconds. From its confusion matrix, fig. 5.9a, it is clear that the model has predicted the two classes just about an equal amount of time, which can also be seen in its specificity score. The same model also had the best result for the **4-class** classifier, though it was very close with the standard EEGNet with 1-second samples. From its confusion matrix fig. 5.9b, it is clear that it prefers to predict "blue" and that it really has trouble with differentiating the different classes.

Model	Rest Vs Task			4 Class	
	Accuracy	F1-Score	Specificity	Accuracy	F1-Score
EEGNet 2s	0.490	0.436	0.804	0.273	0.256
EEGNet 1s	0.497	0.482	0.327	0.274	0.251
Hyp. EEGNet 2s	0.511	0.504	0.4	0.267	0.243
Hyp. EEGNet 1s	0.5	0.373	0.05	0.246	0.166

Table 5.2: The scores on cross subject models, created with data from P2. The score that are highlighted yellow, are the best for that metric. Here Hyp. means that the hyperparameters are found with Hyperband, and 2s and 1s means a sample length of 2 seconds and 1 second, respectively.



(a) **Resting-state vs task** confusion matrix for the model Hyp. EEGNet 2s created with P2 data.

(b) **4-class** confusion matrix for the model EEGNet 2s created with P2 data.

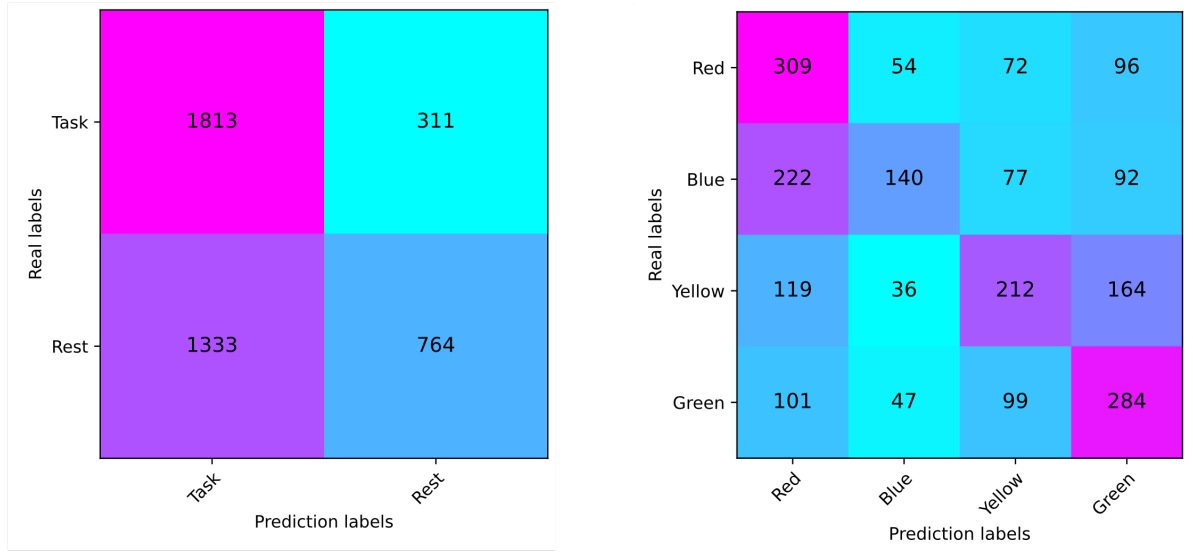
Figure 5.9: Confusion matrices for the best cross subject model for the two classification problems, when the models are trained with P2 data.

5.4.3 Models trained with P1 and P2 data

The result from the different models trained with both P1 and P2 data is presented in table 5.3. The **resting-state vs task** model that performs best is the EEGNet tuned with Hyperband and trained with 2-seconds samples. It can be argued that the standard EEGNet with 2-seconds samples is better than Hyp. EEGNet 2s, since it has higher accuracy and F1. However, as shown in its confusion matrix, fig. 5.10a, the problem with a low specificity means that it predicts "task" too many times, and a BCI that uses such a model will perform "task" when it is not intended. An observation for the **4-class** models is that the Hyperband-based models perform better than the standard EEGNet's. The best performing model was the same as the one in **resting-state vs task**. In its confusion matrix, fig. 5.10b, is it evident that it prefers to predict "red" and "green".

Model	Rest Vs Task			4 Class	
	Accuracy	F1-Score	Specificity	Accuracy	F1-Score
EEGNet 2s	0.611	0.585	0.364	0.402	0.367
EEGNet 1s	0.572	0.522	0.249	0.285	0.233
Hyp. EEGNet 2s	0.573	0.573	0.577	0.445	0.436
Hyp. EEGNet 1s	0.532	0.426	0.103	0.317	0.293

Table 5.3: The scores on cross subject models, created with data from P1 and P2. The score that are highlighted yellow, are the best for that metric. Here Hyp. means that the hyperparameters are found with Hyperband, and 2s and 1s means a sample length of 2 seconds and 1 second, respectively.



(a) **Resting-state vs task** confusion matrix for the model EEGNet 2s created with both P1 and P2 data.

(b) **4-class** confusion matrix for the model Hyp. EEGNet 2s created with both P1 and P2 data.

Figure 5.10: Confusion matrices for the best **4-class** cross subject model and the next best **resting-state vs task** cross subject model, when the models are trained with both P1 and p2 data.

5.5 Cross-subject evaluation with a selection of subjects

Again, the goal of this experiment was to construct a generalized model based on data from several subjects. Though this time, only the subjects that performed well on both classifiers in section 5.1 and section 5.2 were chosen. The reason to use only a selection of the data is to see if the models can be improved further from the ones in section 5.4. Though there is a problem, the pairs that performed best are different based on the data subsets and experiments. Since it is desired to create the best model possible, each data subset uses different selections, which creates a comparison problem, as they all use different subject data. The problem is solved by evaluating their performance with predictions on the entire test set, the same used in section 5.4, and compared against the results from the same experiment. So, there will be two evaluations, one with its own test set and one with the test set from section 5.4. Hyperband search was also used, and similar to the previous experiment in section 5.4, it only had 48 hours to find the best hyperparameters due to time constraints. In the following tables will the acronym SS be used for "subject selection", so the models can be differentiated

from the ones created in the previous experiments.

5.5.1 Models trained with P1 data

With data from P1, eight different classifiers are trained, the same types as in section 5.4. The results in fig. 5.1 and fig. 5.4 showed the different classifier pairs that performed best on the data, implying that the subject selection when the samples are of 2-seconds and 1-second length is different. The selected subjects to create the dataset, are for:

- **Sample length = 2 seconds:** 2, 3, 4, 5, 7, 8, 9, 14, 15, 16, 18, 19, 22, 23, 24, 25, 27, 30, 32, 33 and 34,
- **Sample length = 1 second:** 7, 15, 18, 22, 23 and 33.

Model	Rest Vs Task (Same subjects)			Rest Vs Task (All subjects)		
	Accuracy	F1-Score	Specificity	Accuracy	F1-Score	Specificity
SS EEGNet 2s	0.806	0.802	0.681	0.731	0.720	0.537
SS EEGNet 1s	0.713	0.699	0.499	0.631	0.606	0.380
Hyp. SS EEGNet 2s	0.763	0.762	0.710	0.696	0.694	0.613
Hyp. SS EEGNet 1s	0.692	0.673	0.454	0.630	0.601	0.364

(a) Subject selected, cross subject, model scores when classifying **resting-state vs task** with P1 data.

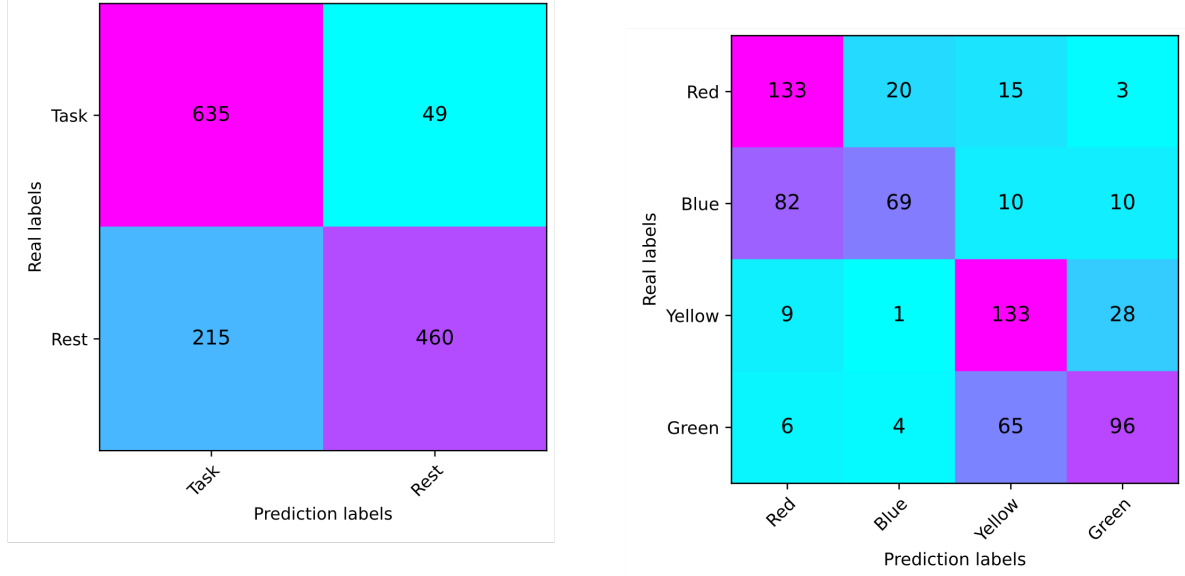
Model	4 Class (Same subjects)		4 Class (All subjects)	
	Accuracy	F1-Score	Accuracy	F1-Score
SS EEGNet 2s	0.611	0.599	0.533	0.491
SS EEGNet 1s	0.488	0.476	0.432	0.401
Hyp. SS EEGNet 2s	0.630	0.621	0.612	0.601
Hyp. SS EEGNet 1s	0.371	0.365	0.405	0.403

(b) Subject selected, cross subject, model scores when classifying **4-class** with P1 data.

Figure 5.11: The scores on cross subject models, created with data from P1. The score that are highlighted yellow, are the best for that metric. The scores highlighted red and green represents improvement or deterioration of the scores compared to the results in table 5.2, respectively. Here Hyp. means that the hyperparameters are found with Hyperband, SS means that the dataset is subject selected, and 2s and 1s means a sample length of 2 seconds and 1 second, respectively.

From the results in fig. 5.11a, it is evident that the models using the 2-seconds samples outperform those using 1-second. In addition, the new standard EEGNet model outperforms the old one when evaluated on its own test set. However, when evaluated on the entire dataset, Hyp. SS EEGNet 2s performs better than SS EEGNet 2s. The high specificity score draws this conclusion. In fig. 5.12a is the confusion matrix to SS EEGNet 2s visualized, and it shows that the models prefer to predict "task" over "rest", and that it is more confident that it is "rest" when predicting "rest".

The results shown in fig. 5.11b shows that, Hyp. SS EEGNet 2s is the best model, though its result is lower than Hyp. EEGNet 2s on both evaluations. From its confusion matrix, fig. 5.12b, is it apparent that it has trouble differentiating between "red" and "blue", and "yellow" and "green".



(a) **Resting-state vs task** confusion matrix for the model SS EEGNet 2s created with subject selected P1 data.

(b) **4-class** confusion matrix for the model Hyp. SS EEGNet 2s created with subject selected P1 data.

Figure 5.12: Confusion matrices for the best cross subject model for the two classification problems, when the models are trained with subject selected P1 data.

5.5.2 Models trained with P2 data

Because of the overall poor performance of the individual models when using a sample length of 1 second, only four models were trained on the P2 data. In other words, the models are only trained with a sample length of 2-seconds, two **resting-state vs task** models and two **4-class** models. Based on the performance of the individual subject models in fig. 5.2, the data from the following subjects are selected:

- 6, 15, 25, and 33.

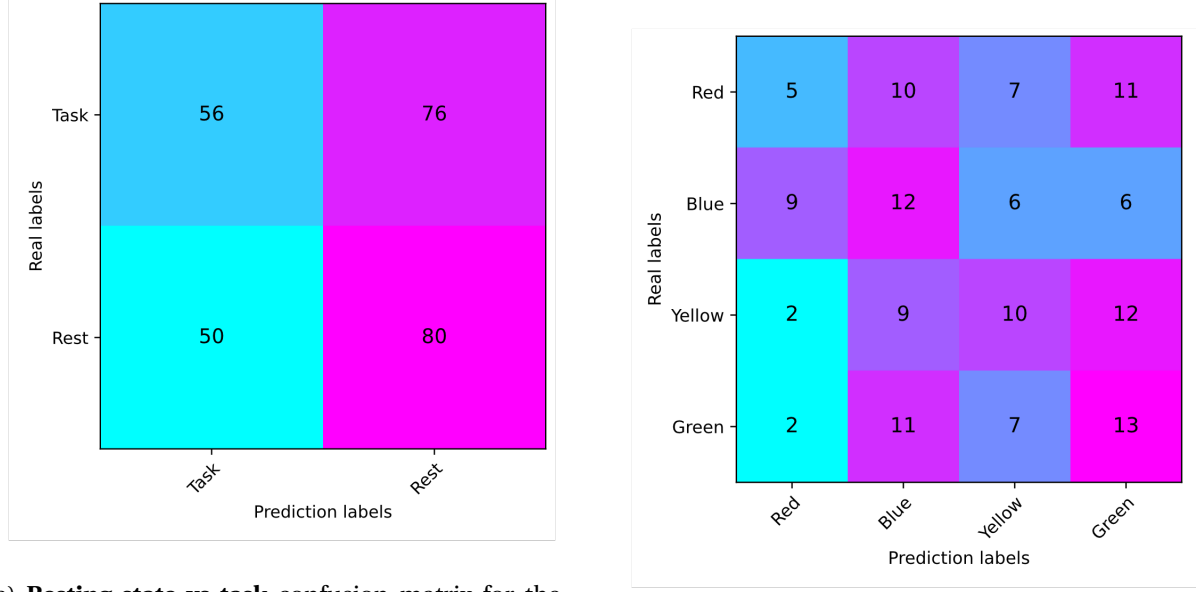
Model	Rest Vs Task (Same subjects)			Rest Vs Task (All subjects)		
	Accuracy	F1-Score	Specificity	Accuracy	F1-Score	Specificity
SS EEGNet 2s	0.519	0.515	0.615	0.515	0.515	0.535
Hyp. SS EEGNet 2s	0.515	0.515	0.492	0.524	0.523	0.493

(a) Subject selected, cross subject, model scores when classifying **resting-state vs task** with P2 data.

Model	4 Class (Same subjects)		4 Class (All subjects)	
	Accuracy	F1-Score	Accuracy	F1-Score
SS EEGNet 2s	0.235	0.22	0.291	0.289
Hyp. SS EEGNet 2s	0.303	0.295	0.277	0.276

(b) Subject selected, cross subject, model scores when classifying **4-class** with P2 data.

Figure 5.13: The scores on cross subject models, created with data from P2. The score that are highlighted yellow, are the best for that metric. The scores highlighted red and green represents improvement or deterioration of the scores compared to the results in table 5.2, respectively. Here Hyp. means that the hyperparameters are found with Hyperband, SS means that the dataset is subject selected, and 2s and 1s means a sample length of 2 seconds and 1 second, respectively.



(a) **Resting-state vs task** confusion matrix for the model SS EEGNet 2s created with subject selected P2 data.

(b) **4-class** confusion matrix for the model Hyp. SS EEGNet 2s created with subject selected P1 data.

Figure 5.14: Confusion matrices for the best cross subject model for the two classification problems, when the models are trained with subject selected P2 data.

Both models in the results in fig. 5.13a manage to perform better than the chance level, though not by much. SS EEGNet 2s are the best of the two and outperform the EEGNet 2s, even though the specificity is now lower when looking at the evaluation done on the entire dataset. Its confusion matrix, viewed in fig. 5.14a, shows that the model prefers to predict "task". Only Hyp. SS EEGNet manages to predict better than chance on its own test set, see fig. 5.13b, though SS EEGNet predicts better on the entire dataset. The confusion matrix to Hyp. SS EEGNet 2s, see fig. 5.14b shows that the model prefers to predict "blue" and "green".

5.5.3 Models trained with P1 and P2 data

Again, because of the poor performance for the individual model created, with P1 and P2 data, when using 1-second samples (see fig. 5.6). The models trained will only be with 2-second samples. In other words, only four models will be trained, two **resting-state vs task** models and two **4-class** models. Based on the performance in fig. 5.3, the data from the following subjects are selected:

- 2, 11, 15, 22, 29, 30, 34.

Based on the results in fig. 5.15a, the model Hyp. SS EEGNet 2s is performing best overall, . When evaluated on its own test set, it performs better than the models in table 5.3. However, it performs worse when evaluated on the entire test set. Its confusion matrix, fig. 5.16a, shows that "task" is the preferred prediction. Hyp. SS EEGNet 2s are also the best model for creating a **4-class** model, see fig. 5.15b. When evaluated on its own test set, it performs better than the models in table 5.3. In addition, when evaluated on the entire test set, the scores are just about the same as the ones to Hyp. EEGNet 2s. In its confusion matrix, fig. 5.16b, it is evident that the model has trouble predicting "yellow", in addition, differentiating between "red" and "blue", and "yellow" and "green".

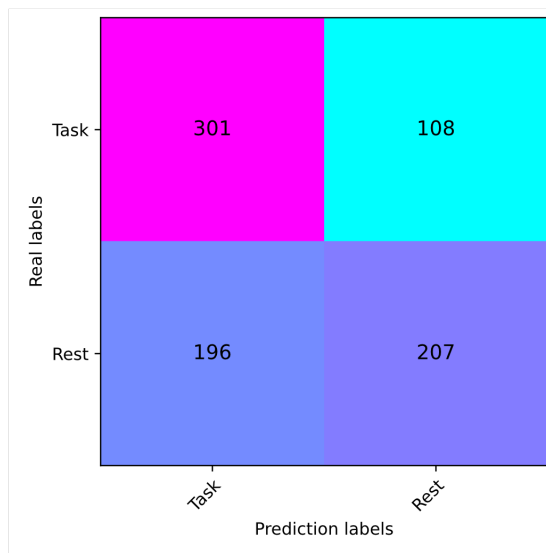
Model	Rest Vs Task (Same subjects)			Rest Vs Task (All subjects)		
	Accuracy	F1-Score	Specificity	Accuracy	F1-Score	Specificity
SS EEGNet 2s	0.628	0.609	0.412	0.593	0.573	0.375
Hyp. SS EEGNet 2s	0.626	0.621	0.514	0.589	0.586	0.509

(a) Subject selected, cross subject, model scores when classifying **resting-state vs task** with both P1 and P2 data.

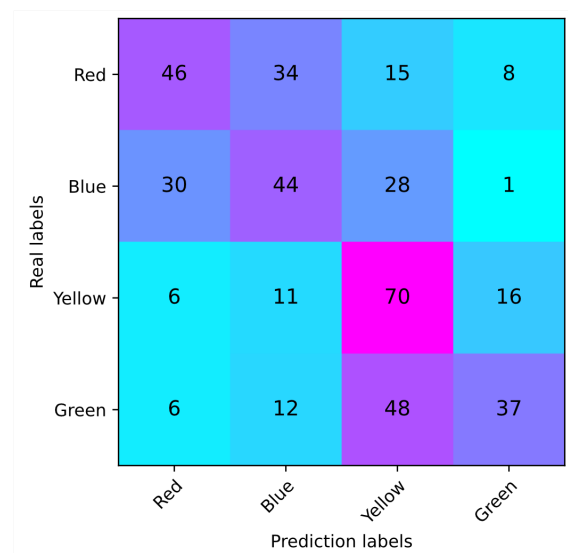
Model	4 Class (Same subjects)		4 Class (All subjects)	
	Accuracy	F1-Score	Accuracy	F1-Score
SS EEGNet 2s	0.464	0.463	0.381	0.341
Hyp. SS EEGNet 2s	0.478	0.473	0.443	0.439

(b) Subject selected, cross subject, model scores when classifying **4-class** with both P1 and P2 data.

Figure 5.15: The scores on cross subject models, created with data from both P1 and P2. The score that are highlighted yellow, are the best for that metric. The scores highlighted red and green represents improvement or deterioration of the scores compared to the results in table 5.3, respectively. Here Hyp. means that the hyperparameters are found with Hyperband, SS means that the dataset is subject selected, and 2s and 1s means a sample length of 2 seconds and 1 second, respectively.



(a) **Resting-state vs task** confusion matrix for the model Hyp. SS EEGNet 2s created with subject selected P1 and P2 data.



(b) **4-class** confusion matrix for the model Hyp. SS EEGNet 2s created with subject selected P1 and P2 data.

Figure 5.16: Confusion matrices for the best cross subject model for the two classification problems, when the models are trained with subject selected P1 and P2 data.

Chapter 6

Discussion and further work

6.1 Discussion

The motivation for this work is to ease the everyday life for persons with LIS and increase their quality of life with the help of a BCI. As mentioned in chapter 1, the design has to be intuitive, accurate and not fatiguing to use over a long time. The tested BCI design only uses eye movement and the brain's color perception to control an external device. This method is not very exhausting compared to motor imagery, and therefore preferably.

Another parameter the BCI should be based on is how fast the BCI can classify the different states. A problem for persons with LIS is that communication is slow and complicated. They can answer a yes/no question fast, but it takes a long time to express sentences, thoughts and feelings. In addition, they are dependent on caregivers to help them with an alphabet board or they can use an eye-tracking device to create words. If not a predefined messages are connected to a option, the classifying speed has to be high, so the user to feel like communicating more or less at the same pace as the interlocutor.

This work consists of two main parts, the data acquisition with the protocols and the classifying of these data with EEGNet. The discussion of each part will be in separate sub-sections below.

6.1.1 Protocols and data acquisition

The testing of the design was done by creating protocols, explained in section 4.3.1, and through data acquisition from 33 different subjects. However, the protocol and the recordings were not optimal, and there is room for improvement to get better data quality.

The first problem was the same constant time period of the "task" and "rest" recordings. The equal period led to the subjects getting used to the rhythmic pace of the recording protocol after a while. Some subjects subconsciously started to breathe at the same pace as the changes in the protocol, like being in a trance, creating rhythmic noise in the data. Another problem with the constant period is the reduction of the subjects' vigilance. Their eye movement and focus would, after a while be automatic and subconscious, no longer looking at the states with focus. Several subjects also noted that they moved their focus to the "rest" state in the middle of the screen too early, as they knew a resting-state would follow soon.

A solution to the problems described above is to randomize the length of the states. By doing this the rhythm of the protocol would disappear and increase the attention span of the subjects as they do not know precisely when the state will change. The reason for not implementing this in the protocol is that the input to the classifier has to be of equal length. However, it would be possible to resample the data so they all will be of the same length, though this would have removed available data. Another method would have been to use a feature extraction process on the data and use those features as input to the CNN. The order of "task" and "rest" state can not be randomized, as the desired movement the classifier shall learn are from "rest" to "task", not "task" to "task" or "rest" to "rest".

A problem with the execution of the data acquisition was the limited amount of metadata. The only metadata collected, except what can be found by looking at the dataset, where if the subject had epilepsy or was color blind. Additional information should have been collected, such as asking how tired the subject was before the data gathering started, how exhausting the protocol was, and if there were some sessions in which they became unconcentrated. The answers to these questions could help remove the sessions that might have been too noisy because of an unconcentrated or exhausted subject.

A complication with the dataset is that the amounts of data points are different in each session. Simultaneously, the sample rate and marks on all sessions are identical, as explained in section 4.3.2. The problem occurs when splitting a session into samples corresponding to a specific state. There will be a difference between the actual state and where the marks say it is. Marks corresponding to a (e.g., a resting state) have mixed data from the "rest"- and "task" state, which is not ideal. In other words, the different amounts of data points can have a directly effect on the experiments' results in chapter 5. In addition, it is challenging to find when or where the shifts happened. So when the sessions were partitioned into samples, the length of the session was not taken into account.

There was also a two-second difference between P1 and P2, seen in fig. 4.6. P1 has the correct number of data points, while P2 recorded 4 seconds more. If the extra data in P2 was caused by the software loading the protocol and images at the start, then this is not a large issue. Because P2 has only two states, rest or task, simply put. These states will always follow after one another, and a shift of 4 seconds has no effect, other than that the four first samples are wrong. Since the first sample is consistently removed, only three samples is wrong, which is a minor error. They were therefore all kept in the final version of the dataset, as it is unknown if the shift happened at the start.

The recording environment introduced minimal non-physiological artifacts, implying that the recorded signal was much cleaner than if it was recorded in the real world. Possible new artifacts in the real world are sound noise, movement, and specific for persons with LIS, noise from their life-giving medical equipment. Even though the signal is being pre-processed, it is still unknown if the processed signal is as clean as the one gathered in the laboratory. It is important, as mentioned in chapter 3, that the BCI can withstand real-life noise. If not, it is of little value.

6.1.2 Classifiers

Common for all the conducted experiments was that the data was pre-processed before creating the classifiers. The sole reason to pre-process the raw data is to remove artifacts and noise, making it easier to classify the samples. In this work, the whole session is being pre-processed simultaneously, which takes some time, and be a challenge to real-time implementation. Since pre-processing costs both time and computational power, it may be better to let the EEGNet create its own processing filters in its hidden layers, as was done by its creators [20]. Even though this work utilized pre-processing, no experiments were conducted to find the best pre-processing methods for the data.

Using a deep learning method to classify EEG data has some problems, compared to the other well-performing ML classifiers mentioned in chapter 3. A CNN needs a lot of computational power and time to train, before it is ready. These issues are not problematic in an offline setting, but can be in a real-time setting, if the model needs to recalibrate and is placed on a limited microcontroller. The calibration may take a long time which is not optimal, implying that a user can have trouble utilizing the BCI's full effect. Another problem with deep learning algorithms are the difficulties in understanding how a prediction is made. In study [20], they use three different methods to characterize the features learned by the model. This visualization could have been implemented to understand why the models have the performance they do.

A general observation when evaluating all the different results in chapter 5, is that the models created with P1 are performing much better than the models created with the other subsets of the data. The difference between P1 and P2 is that P2 has an icon on each colored box, making the boxes labeled, implying that the user knows what the state does. If there are P1 states (as presented in fig. 4.3a), the user needs to remember the meaning of each color. Increasing the BCI's difficulty level makes it no longer as effective and convenient as the one with P2 states. Another observation is that **resting-state vs task** classifiers generally do better than the **4-class** classifiers. A reason for this is that the "rest" samples have a lot more blinking in them, as the subjects were asked to blink when the "rest" state was showing if needed. Blinks are an artifact that is very recognizable in the raw EEG signals since it gets much higher amplitudes because of muscle contractions. The models may have found blinking to reference the "rest" state.

Only the models created with P1 data performed well in the experiments in section 5.1. Models with a performance close to the chance line, implies that they have not learned anything from the training data. In addition, for the **resting-state vs task** models, when the models have low specificity, or very high, and the accuracy and F1 is around 0.5, the model has overfitted and learned to only predict one of the variables, which is not desired.

The experiments in section 5.2 have results very similar to the ones in section 5.1. Only with the P1 data subset several classifier pairs got an average good performance. Again, low specificity is a problem for the **resting-state vs task** models, implying that it might have been beneficial to use class weights when training, so the models would prefer to predict "rest" instead of "task". Since it is more irritating and even dangerous that the BCI performs random commands, which would be the case

when wrongly predicting "task", than just being slow when trying to do a "task". The **4-class** models were overall better than the **resting-state vs task** models, though they did not perform as good as the ones created with 2-seconds samples. The goal for this experiment was to see if the accuracy could be kept high if sample length were lowered. The results is clear: only the models created with P1, are possible. However, there may be other optimization techniques or ML methods that could increase the performance.

As previously discussed, there is an advantage of using the states in P2, though the models created with P2 have not been performing well. The experiments from section 5.3 showed that it was no performance to gain by using models created with P1 data to predict P2 data. Only one model, the **4-class** model for subject 7, showed a significant increase. However, this was most likely "luck" since none of the other models had a similar increase. What could have improved the results was to use the P1 models as a base and continue to train the weights with P2 data before evaluating them on the P2 test set.

Again the experiments in section 5.4 show that **resting-state vs task** models struggled with getting a high specificity. So using class weights could improve the performance of the models. In addition, using samples with the original length, 2-seconds, is superior to using a shorter sample length. Models created with P1 data and 1-second samples had accuracy, and F1 score close to models trained with longer samples. In other words, it might be possible to shorten the length of the samples without decreasing the accuracy, which is relevant for real-time implementation. Only models created with data from P1, and both P1 and P2 performed adequately. The models created with P1 data still performed the best. From the confusion matrix, fig. 5.8b, it is evident that the best **4-class** model has problems differentiating between "red" and "blue", and between "yellow" and "green". The "red" and "blue" states were placed on the left side of the screen, while the other two were placed on the right side. The problem for the model to differentiate within these groups, implies that the electrodes might struggle with reading vertical eye movement. This hypothesis can be checked by training models only on first, e.g., "red" and "blue" samples, then "red" and "yellow" samples.

The experiments in section 5.5 showed that the model performances could be further improved by only training with a subset of subject data. Furthermore, the models trained with P1 data have improved the most. A main thing to note is that the specificity is now generally higher than the results from the experiments in section 5.4. Though, preferably it should have been even closer to 1. In the confusion matrix (fig. 5.12b) is it even more evident that the model has problems differentiating between "red" and "blue", and between "yellow" and "green", this strengthens the hypothesis that the electrode has problems with collecting and reading vertical eye movement.

The results from the experiments in section 5.4 and section 5.5 showed that the standard hyperparameters to EEGNet were usually better than the ones found with the Hyperband search. A reason behind is that the Hyperband did not finish its search, i.e., it did not converge to the best hyperparameters. Another reason is that Hyperband maximized the validation accuracy, which led to its best models suffering from overfitting. By optimizing the validation loss instead, the chance of overfitting could have decreased. The parameters found with Hyperband improved the **4-class** models the most

and usually helped **resting-state vs task** models increase their specificity.

To summarize, the best models were created with data from P1 and a sample length of 2-seconds. The highest accuracy created with cross-subject data were 80.6% (see fig. 5.11a) on a resting-state vs task model and 65.1% (see table 5.1) on a 4-class model. Though the experiments did not focus on pre-processing, there may be better pre-processing methods that would have improved performance on the different models. Subject selection on the dataset and using other hyperparameters than the default ones, increased the performance of models based on 1-second samples. Making it possible to predict on shorter samples and increasing the speed of the BCI, though more testing is needed before this would work perfectly. The **resting-state vs task** models generally predicted a low specificity, which can be improved by using class weights when training. The Hyperband search also showed the best improvement on **4-class** models, implying that these models could further improve with the correct hyperparameters.

6.2 Further work

The next step for this work is to implement the BCI for real-time use and examine how it performs in an online environment. However, before a real-time application can be made, the BCI design should be evaluated more thoroughly, and a protocol must be determined, either one that were used in this work, P1 or P2, or a completely new one. Evaluating the BCI design can be done by discussing it with persons with LIS and caregivers, listening to their needs and preferences. In addition it is essential to understand *when* and *for what* the BCI should be relevant to use for persons with LIS. It is also important to note that current BCI applications are only used for short periods at a time.

Implementing a real-time application will also require the design of a state machine. The state machine will decide what the different tasks do and how it is possible to switch between tasks. In addition, based on the results in this work, a real-time application will need to contain two different classifiers, one that differentiates between "rest" and "task", and one that differentiates the tasks. If the real-time application has to be designed equal to the one in this work, then a choice between the two different protocol states has to be made. P1 yielded the best results, though it requires the user to remember what each color means. There may be ways to design the viewing screen such that P1 can be used, by, e.g., having a description below the colored boxes or by splitting the screen, where one part is the states, and the other is an explanation.

There are some challenges with real-time applications that need to be solved. First: there is no definition to when a task is starting, the BCI has to understand when the user wants to perform a task. Next: to what degree shall the user be able to do something else with its brain? In other words, should the user be able to focus and think on something completely different than the control of the BCI while connected to it? Thirdly: how to implement the classifiers. Are they untrained before using the BCI, trained but need to be adapted with new user data, recorded right before use with the same protocol, or have they finished training and are ready to use. Finally: how long time is maximum for the classifiers to recalibrate and train new models with updated or new EEG signals. The IDUN cluster will not be available in a real-time setting.

Apart from creating a real-time application of the BCI, several experiences and tests can be performed on the dataset. To speed up the BCI while keeping its accuracy, one might, e.g., work to get good performance with as few data points as possible. One could also remove the EEG signals generated from the channels above the eyes and get the best result possible by testing the availability of the BCI for persons with CLIS. Tests to find the culprit to the extra data points in P2, should also be conducted, as the answer may improve the performance of models trained with P2 data. Feature extraction methods, such as discrete wavelet transform and empirical mode decomposition, can also be implemented to see if the extraction will help the model's performance. The generalized model's performances should also be tested to check if they can identify EEG signals from new subjects.

References

- [1] Taras Halan, Juan Fernando Ortiz, Dinesh Reddy, Abbas Altamimi, Abimbola O Ajibowo, and Stephanie P Fabara. [Locked-In Syndrome: A Systematic Review of Long-Term Management and Prognosis](#). *Cureus*, 13(7), 2021.
- [2] Veronica Johansson, Surjo R Soekadar, and Jens Clausen. [Locked out: Ignorance and responsibility in brain–computer interface communication in locked-in syndrome](#). *Cambridge Quarterly of Healthcare Ethics*, 26(4):555–576, 2017.
- [3] Marie-Aur lie Bruno, Jan L Bernheim, Didier Ledoux, Fr d ric Pellas, Athena Demertzi, and Steven Laureys. [A survey on self-assessed well-being in a cohort of chronic locked-in syndrome patients: happy majority, miserable minority](#). *BMJ open*, 1(1):e000039, 2011.
- [4] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. [Brain–computer interfaces for communication and control](#). *Clinical neurophysiology*, 113(6):767–791, 2002.
- [5] Xiaorong Gao, Yijun Wang, Xiaogang Chen, and Shangkai Gao. [Interface, interaction, and intelligence in generalized brain–computer interfaces](#). *Trends in Cognitive Sciences*, 2021.
- [6] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. [Brain computer interfaces, a review](#). *sensors*, 12(2):1211–1279, 2012.
- [7] Sunnaas sykehus. Locked-in syndrom. <https://www.sunnaas.no/behandlinger/locked-in-syndrom#helsepersonell-sjekkliste-for-henvisning---fastlege-eller-annen-helsetjeneste-henviser-til-utredning>, (n.d.). [Online; accessed 12-December-2021].
- [8] Johns Hopkins Medicine. Brain anatomy and how the brain works. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/anatomy-of-the-brain>, journal=Johns Hopkins Medicine, (n.d.). [Online; accessed 14-December-2021].
- [9] D. Purves, D. Fitzpatrick, L.C. Katz, A.S. Lamantia, J.O. McNamara, S.M. Williams, and G.J. Augustine. *Neuroscience*. Sinauer Associates, 5 edition, 2011.
- [10] John H. Martin. *The Visual System*. McGraw Hill, New York, NY, 2021.
- [11] Per Brodal. *Sentralnervesystemet*. Universitetsforlaget, Oslo, 5 edition, 2012.
- [12] In es Bram o, Lu s Fa sca, Christian Forkstam, Alexandra Reis, and Karl Magnus Petersson. [Cortical brain regions associated with color processing: An FMRI study](#). *The open neuroimaging journal*, 4:164, 2010.

- [13] St Louis Erik K., Lauren C. Frey, and Jeffrey W. Britton. *Electroencephalography (EEG): An introductory text and Atlas of Normal and abnormal findings in adults, children, and infants*. American Epilepsy Society, 2016.
- [14] Luis Alfredo Moctezuma. *Towards Universal EEG systems with minimum channel count based on Machine Learning and Computational Intelligence*. PhD thesis, Norwegian University of Science and Technology, 2021.
- [15] Li Hu and Zhiguo Zhang. *EEG Signal Processing and Feature Extraction*. Springer Singapore Pte. Limited, Singapore, 2019.
- [16] William O Tatum, Barbara A Dworetzky, and Donald L Schomer. [Artifact and recording concepts in EEG](#). *Journal of clinical neurophysiology*, 28(3):252–263, 2011.
- [17] Jinhu Xiong, Liangsuo Ma, Binqun Wang, Shalini Narayana, Eugene P Duff, Gary F Egan, and Peter T Fox. [Long-term motor training induced changes in regional cerebral blood flow in both task and resting states](#). *Neuroimage*, 45(1):75–82, 2009.
- [18] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. [A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update](#). *Journal of neural engineering*, 15(3):031005, 2018.
- [19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [20] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. [EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces](#). *Journal of neural engineering*, 15(5):056013, 2018.
- [21] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. [Hyperband: A novel bandit-based approach to hyperparameter optimization](#). *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.
- [22] Mariana P Branco, Elmar GM Pels, Femke Nijboer, Nick F Ramsey, and Mariska J Vansteensel. [Brain-Computer interfaces for communication: preferences of individuals with locked-in syndrome, caregivers and researchers](#). *Disability and Rehabilitation: Assistive Technology*, pages 1–11, 2021.
- [23] Wessam Shehieb, Sara Alansari, and Nada Jadallah. [EEG-based communication system for patients with locked-in syndrome using fuzzy logic](#). In *2017 10th Biomedical Engineering International Conference (BMEiCON)*, pages 1–5. IEEE, 2017.
- [24] Han-Jeong Hwang, Kiwoon Kwon, and Chang-Hwang Im. [Neurofeedback-based motor imagery training for brain–computer interface \(BCI\)](#). *Journal of neuroscience methods*, 179(1):150–156, 2009.
- [25] Upasana Talukdar, Shyamanta M Hazarika, and John Q Gan. [Adaptation of Common Spatial Patterns based on mental fatigue for motor-imagery BCI](#). *Biomedical Signal Processing and Control*, 58:101829, 2020.

- [26] Upasana Talukdar, Shyamanta M Hazarika, and John Q Gan. [Motor imagery and mental fatigue: inter-relationship and EEG based estimation](#). *Journal of computational neuroscience*, 46(1):55–76, 2019.
- [27] Alessandro Tonin, Andres Jaramillo-Gonzalez, Aygul Rana, Majid Khalili-Ardali, Niels Birbaumer, and Ujwal Chaudhary. [Auditory electrooculogram-based communication system for ALS patients in transition from locked-in to complete locked-in state](#). *Scientific reports*, 10(1):1–10, 2020.
- [28] Chang-Hee Han, Yong-Wook Kim, Seung Hyun Kim, Zoran Nenadic, Chang-Hwan Im, et al. [Electroencephalography-based endogenous brain–computer interface for online communication with a completely locked-in patient](#). *Journal of neuroengineering and rehabilitation*, 16(1):1–13, 2019.
- [29] Evangelos Antoniou, Pavlos Bozios, Vasileios Christou, Katerina D Tzimourta, Konstantinos Kalafatakis, Markos G Tsipouras, Nikolaos Giannakeas, and Alexandros T Tzallas. [EEG-Based Eye Movement Recognition Using the Brain–Computer Interface and Random Forests](#). *Sensors*, 21(7):2339, 2021.
- [30] Robin Schirrmeister, Jost Springenberg, Lukas Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human eeg. 03 2017.
- [31] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. [Generative adversarial nets](#). *Advances in neural information processing systems*, 27, 2014.
- [32] OpenBCI Online Store. Ultracortex 'mark iv' eeg headset. <https://shop.openbci.com/collections/frontpage/products/ultracortex-mark-iv?variant=23280741699>, (n.d.). [Online; accessed 14-December-2021].
- [33] Magnus Sjölander, Magnus Jahre, Gunnar Tufte, and Nico Reissmann. [EPIC: An Energy-Efficient, High-Performance GPGPU Computing Research Infrastructure](#), 2019.