Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

CARLOS GONZALEZ VISIEDO

Deep Learning EEG-based Motor Imagery System for Robot Control using 3D Printed Headset and Electrodes

DEGREE PROGRAMME IN DATA ENGINEERING
2023

Author(s) Gonzalez Visiedo, Carlos	Type of Publication Bachelor’s thesis	Date June 2023
	Number of pages 108	Language of publication: English

Title of publication

Deep Learning EEG-based Motor Imagery System for Robot Control using 3D Printed Headset and Electrodes

Degree Programme

Data Engineering – Bachelor of Engineering

Abstract:

This thesis describes the design and implementation of an EEG-based motor imagery system for robot control using a 3D printed headset and electrodes. The primary aim was to create a more comfortable and user-friendly EEG headset that, in combination with a deep learning model, can reliably measure EEG signal and classify motor imagery for controlling a robot arm.

Research in this area involves 3D scanning and printing a human head to design and fabricate a custom EEG headset with integrated detachable electrodes. Different electrode materials and coatings were evaluated to determine which ones were most suitable for EEG signal measurement compared with commercial electrodes. Furthermore, machine learning models for binary EEG signal classification using CNNs and transfer learning were developed.

The trained model with best accuracy was then integrated with ROS MoveIt package for controlling a robot arm using user's motor imagery EEG signals. Results showed that the developed EEG headset and electrodes provided reliable, accurate EEG signal measurements for robot control. CNN models achieved high classification accuracy of 93% on public dataset, but poor generalization on personal dataset.

Transfer learning provided similar accuracy in comparison with the models trained on public dataset while significantly improved the performance of the model on personal dataset. In overall, the best CNN model achieved average accuracy rate of 62.5% when testing was made with EEG data obtained in different environments. By connecting the machine learning model to ROS MoveIt package, specific predefined movements can be executed based on user's motor imagery EEG signals. Overall, this research presents a promising path towards creating more comfortable and effective EEG-based robot control systems.

Key words

Electroencephalography, EEG, EEG sensors, brain-computer interfaces, BCIs, 3D printing, data analysis, machine learning, deep learning, transfer learning, robotics

FOREWORD

I am delighted to present this thesis, which is the result of my personal efforts and commitment towards developing an EEG-based motor imagery system for robot control.

I wish to acknowledge the crucial role played by RoboAI in the successful completion of my research. The resources and instruments provided by the company were essential in enabling me to carry out my work to a high standard. I am grateful for the trust and support shown to me by the company, which has allowed me to undertake such an exciting and challenging project. I am proud to have been associated with such a supportive and forward-thinking organization.

Throughout this research, I encountered various challenges that required me to use my creativity and problem-solving skills to overcome. I am grateful for the support and expertise of some specific workmates from my company, such as Joonas Kortelainen, who helped me with 3D printing techniques, and my colleagues who provided valuable feedback on my work.

I would also like to express my sincere gratitude to all the teachers of my degree program who have provided me with the necessary knowledge and skills to conduct this research. I am particularly grateful to my thesis coordinator Toni Aaltonen and my tutor Petteri Pulkkinen for their guidance and support throughout the entire process.

Finally, I would like to acknowledge the support of my friends, who provided me with emotional support and encouragement throughout this challenging journey.

I hope that the results of this research will contribute to the development of more comfortable and effective EEG-based motor imagery systems for robot control, and I invite readers to explore this thesis further.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND TERMS

BCI – Brain Computer Interfaces

CAR – Common Average Reference

CNN – Convolutional Neural Network

CSP – Common Spatial Pattern

DFT – Discrete Fourier Transform

DWT – Discrete Wavelet Transform

ECG – Electrocardiography

EEG – Electroencephalography

EMG – Electromyography

FN – False Negatives

FP – False Positives

ICA – Independent Component Analysis

LSL – Lab Streaming Layer

MI – Motor Imagery

OpenBCI – Open-Source Brain-Computer Interface

ROS – Robot Operating System

RNN – Recurrent Neural Network

Rviz – ROS Visualization Tool

TF – Time-Frequency

TN – True Negatives

TP – True Positives

1 INTRODUCTION

The use of brain-computer interfaces (BCIs) has gained significant interest in recent years due to their potential to revolutionize the way humans interact with technology. BCIs provide a direct communication route between our brain and an external device, enabling those with physical disabilities to control computers or robots using just their thoughts (Värbu et al., 2022, pp. 1-2). Electroencephalography signals (EEG) signals have recently become an area of emphasis within BCI research. One specific application could be in controlling robotic devices.

This thesis seeks to investigate the use of EEG signals for robot control by taking a novel approach involving 3D printing technology for custom EEG headset and electrode fabrication. This study utilized BCI Competition IV - Graz Data Set A as a publicly available dataset in order to develop and assess machine learning algorithms for EEG signal classification.

Chapter 1 presents an overview of the motivation and background for this research study, its problem statement, research questions, objectives and scope as well as its contribution and significance. Chapter 2 contains an in-depth literature review covering motor imagery and EEG research including BCIs, EEG headset design incorporating electrode design for BCIs as well as data acquisition, preprocessing methods and machine learning approaches for EEG signal classification as well as transfer learning. Chapter 3 details methodology employed during this investigation including experimental designs as well as EEG headset, electrode design and fabrication, data acquisition, preprocessing methods, machine learning algorithms and transfer learning approaches used during this investigation. Chapter 4 presents results of the experiment including EEG headset performance with electrodes attached, CNN model classification accuracy and integration with robot arm. Chapter 5 discusses implications, limitations and suggests future research initiatives; and Chapter 6 concludes the thesis with final thoughts.

The findings of this study have the potential to contribute to the development of more effective and accessible BCIs that can be used to improve the quality of life for individuals with physical disabilities.

1.1 Background and Motivation

EEG is a non-invasive technique to measure electrical activity in the brain. One application of EEG is motor imagery (MI), where individuals visualize performing specific movements to generate commands for robots. This approach could have applications in rehabilitation (Ang & Guan, 2013, pp. 141-144) and assistive technologies for individuals with motor impairments.

However, current approaches to EEG-based MI for robot control have limitations, such as the need to wear commercial EEG headsets and electrodes that may be expensive and uncomfortable to users. Furthermore, machine learning models used for MI classification are not performing well with multiple classification tasks directly because Common Spatial Pattern (CSP) is a binary feature extraction method (An et al., 2023, p. 12004). If CSP is used for multi-classification tasks, some other strategies must be used.

This research seeks to address these limitations and enhance user experience and performance when using EEG-based MI for robot control. To do so, this thesis proposes designing and 3D printing a custom EEG headset and electrodes that improve comfort during measurements while achieving similar accuracy of EEG signal measurements in comparison to commercial devices. Furthermore, machine learning algorithms such as Convolutional Neural Networks (CNNs) will be employed in MI classification as well as transfer learning to enhance model performance and generalization.

1.2 Problem Statement

Previous studies have demonstrated promising results when applying EEG-based MI to robot control (Herath & deMel, 2021, pp. 7-14), however there are still caveats that need to be addressed. One major drawback is the use of commercial EEG headsets and electrodes which may be uncomfortable or tedious to wear. Furthermore, machine learning models used may require extensive training and calibration to achieve optimal performance, which can be time-consuming and resource-intensive. Additionally, the accuracy and reliability of MI-based robot control can be affected by factors such as fatigue, distractions, and changes in brain activity patterns over time (Liu et al., 2022, pp. 269-272).

This thesis seeks to solve two problems: designing and 3D printing a customized EEG headset and electrodes for improved user experience, obtaining good accuracy of EEG signal during measurements, and employing machine learning algorithms such as CNNs for MI classification and transfer learning to enhance model performance and generalization.

The proposed solution to this problem is to design and 3D print a custom EEG headset and electrodes that are comfortable yet accurately capture EEG signals. Furthermore, multiple CNN models will be trained and evaluated for classification of MI, while transfer learning will be implemented with the aim of improving performance and generalization of machine learning models. Finally, integration between the machine learning model and ROS MoveIt package will enable control of a robotic arm, testing its feasibility in real-world applications.

The significance of this issue lies in its potential applications in robotics and in the health field such rehabilitation and assistive technologies for individuals with motor impairments. Improving EEG measurements and machine learning models to make them more accurate and user friendly could result in the development of more EEG-based controlled devices and therefore in more accessible rehabilitation and assistive technologies tailored towards this population.

1.3 Research Questions

This thesis seeks to address the problem statement outlined above by answering the following research questions:

1. What is the optimal design and 3D printing process for a custom EEG headset and electrodes that provide accurate, comfortable measurements of EEG signal?
2. How can machine learning algorithms, including CNNs, be employed to classify MI and generalize them to new EEG signal data?
3. Can transfer learning improve the performance and generalization of machine learning models for MI classification?
4. How can the machine learning model be utilized to control a robot arm and perform specific movements both virtually, as well as with real robots?
5. What are the potential advantages and drawbacks of EEG-based MI for controlling robots and other devices?

These research questions will inform the design, implementation and evaluation of any proposed solution to this problem statement.

1.4 Objectives and Scope

This thesis seeks to design and develop a custom EEG headset and electrodes for accurate, comfortable EEG signal measurements during MI, as well as employ machine learning algorithms to classify MI for robot control applications. The specific objectives include:

1. To design and 3D print a custom EEG headset and electrodes that offer improved comfort over existing commercial products.
2. Acquire and preprocess EEG signal data for MI classification, including feature extraction and time-frequency (TF) analysis.
3. To train and assess multiple machine learning models, including CNNs for MI classification using both public datasets as well as newly collected EEG measurements data.
4. To assess the feasibility of transfer learning to enhance machine learning models used for MI classification.

5. Utilizing the machine learning model combined with the ROS MoveIt package to program a robot arm to perform specific movements both virtually in simulation and/or with an actual robot.
6. To assess the feasibility and effectiveness of EEG-based Motor Input for Robot Control.

This thesis examines the design and development of a custom EEG headset and electrodes, as well as classification of MI for robot control using machine learning algorithms.

1.5 Contribution and Significance

This thesis presents the development of a custom EEG headset and electrodes for accurate, comfortable EEG signal measurements during MI. The design, 3D printing process, characteristics, and performance measurements are all discussed in detail.

Another important contribution is the application of machine learning algorithms, including CNNs, to classify MI for robot control. Multiple models are trained and evaluated with both public datasets and newly collected data with the goal of improving model performance and generalization through transfer learning.

This research is significant because it explores the potential applications of EEG-based MI for controlling robots and other devices. It provides an understanding of the feasibility and effectiveness of using MI as a control strategy for robotic devices, while emphasizing the need to develop custom EEG measurement systems that are accurate, comfortable, and cost-effective.

This thesis seeks to make a contribution to the field of BCIs by optimizing EEG-based MI for robot control, as well as exploring its potential applications.

2 LITERATURE REVIEW

2.1 Motor Imagery and EEG

MI is a mental process where someone imagines performing an action without actually carrying it out. According to Guillot et al. (2012, p. 2), MI activates similar neural circuits as actual movement but in a lesser extent as actual movement, and thus can be used as an indirect measure of motor activity. EEG is a non-invasive recording technique of brain electrical activity using electrodes placed on the scalp. EEG can detect changes associated with MI, making it an invaluable tool in unravelling motor control's neural correlates.

Studies have investigated the use of EEG for decoding MI, with the ultimate goal of developing BCIs that allow individuals to control external devices with their brain activity (Saibene et al., 2022, pp. 1-2). MI-BCIs in particular have shown promise in controlling robotic devices like robotic arms for uses like rehabilitation and assistive technologies applications (Onose et al., 2012, pp. 604-607).

MI is believed to be caused by activation of the sensorimotor cortex, which is responsible for motor planning and execution. When MI occurs, this area generates neural oscillations in alpha and beta frequency bands that can be detected using EEG. Analysis of these oscillations provides information regarding motivation to move, direction of movement, and timing of movements. (Kim et al., 2016, pp. 2-11.)

EEG-based MI-BCIs for robotic control typically involve recording EEG signals during MI tasks and classifying these into different movement classes using machine learning algorithms. The output of this machine learning algorithm is then utilized to control movement of a robotic device such as an arm using an appropriate robotic control system such as ROS MoveIt package. (Padfield et al., 2019.)

In the following sections, it will assess the existing literature on EEG-based MI and machine learning algorithms for MI classification, as well as discuss potential applications of EEG-based MI to robot control.

2.2 Brain-Computer Interfaces (BCI’s)

BCIs are an assistive technology that enables individuals to communicate with external devices by using their brain activity alone, without needing any physical movements. BCIs can be employed for various applications such as communication and robotic device control. (Kulkarni & Bhosale, 2018, p. 62.)

BCIs can be divided into distinct types based on the brain signals they use for communication. EEG-based BCIs use EEG signals recorded from the scalp to detect changes in brain activity associated with motor imagery or other cognitive processes;

magnetoencephalography (MEG)-based BCIs use magnetic signals generated by the brain to detect changes in activity; while invasive BCIs utilize implanted electrodes to record direct from cortex activity. (Rak et al., 2012, pp. 430-431.)

EEG-based BCIs are the most commonly used type of BCI, due to their noninvasive nature and relatively low cost. EEG-based BCIs can be employed for many applications such as communication or control of external devices. (Jamil et al., 2021, pp. 6-9.)

EEG-based BCIs typically involve the recording of EEG signals during motor imagery or other cognitive tasks, followed by classification using machine learning algorithms. The output from this machine learning algorithm is then used to control the movement of a robotic device such as an arm.

In the following sections, it will be reviewed the existing literature on EEG-based BCIs, including different types of signals used for communication, machine learning algorithms used for signal classification and potential applications of EEG-based BCIs for robotic control.

2.3 EEG Headset and Electrode Design

For robot control applications, high-quality EEG signals require the use of an ergonomic headset and electrode system that provides comfort, is user friendly, and records accurate signals with ease (Muhammad et al., 2023, pp. 2822-2831). Unfortunately, many commercial EEG headsets and electrodes can be uncomfortable or cumbersome to use, leading to skin irritation or discomfort for the user.

3D scanning techniques can be used to create personalized and anatomically fabricated EEG headsets, enabling fit customization. This optimization improves user comfort and ensures accurate electrode placement, resulting in more reliable EEG measurements. The direct integration of electrodes into the structure of the 3D-printed EEG headset eliminates the need for separate adhesive or gel electrodes. This combination increases functionality and simplifies electrode placement. (Lacko et al., 2016, pp. 129-130.) 3D printing enables rapid prototyping, allowing modifications based on user feedback. This process improves the user experience and improves the accuracy of the EEG measurement since it ensures a better contact between the electrodes and the scalp. However, challenges remain, especially in terms of longevity and durability of 3D printed materials. Careful consideration must be given to material choices and construction techniques to ensure the structural integrity and long-term drainage of the structure.

Impedance and signal quality tests are required to ensure the 3D printed EEG headset electrodes are effective and reliable. These tests measure the conductivity of the printed electrode and check the quality of the acquired EEG signals. Researchers use a variety of methods to measure impedance, such as using an impedance meter or analyzing the output voltage using a known test signal (Sohal et al., 2018, pp. 98-

101). Despite that Koctúrová & Juhár (2019, p. 4) found that resistance does not directly affect EEG signal quality, resistance can still be measured to obtain important information about the conductivity of the electrodes. The signal quality is analyzed to verify the accuracy and validity of the acquired EEG signal. These tests examine noise levels, signal-to-noise ratios, and artifacts. High-quality signals are necessary for accurate interpretation and analysis of brain activity. Signal quality testing involves the subject remaining still and relaxed, as well as recording EEG signals during controlled activities or specific cognitive tasks and then analyzing the recorded signals for noise, whether artifacts, and the overall quality of the signal (Weber et al., 2021, pp. 1-4). Researchers are comparing the signal properties of 3D printed electrodes with conventional electrodes to determine their suitability for EEG applications.

It is crucial to carefully select and test electrodes to ensure optimal signal quality in EEG studies. Electrode impedance is a critical parameter in the performance of EEG electrodes, as it directly influences both quality and accuracy of recorded signals (Rahman et al., 2022, p. 1879). Designs, materials and coatings of EEG electrodes all play an influential role in their impedance characteristics.

2.4 Data Acquisition

Data acquisition and preprocessing are critical steps in the analysis of EEG signals for BCI applications. Numerous studies have been done to examine different approaches to data acquisition and preprocessing in order to guarantee reliable and accurate analysis (Puce & Hämäläinen, 2017, pp. 8-21).

EEG data acquisition involves using EEG recording equipment to record electrical signals generated in the brain with scalp electrodes. The quality of the data can be greatly affected by the EEG recording system used, the electrode placement and the number of electrodes used (J. Montoya-Martínez et al., 2021, pp. 7-16). There are many different types of EEG recording system options, including wired or wireless systems and dry or wet electrodes. Also, there are various electrode montages (e.g., 10-20 system, 10-10, etc.). All of these systems have their advantages and disadvantages (Acharya & Acharya, 2019, pp. 326-328). The specific research needs and experimental design are key factors in determining the right EEG recording device Mitocaru et al. (2021, pp. 192, 195).

2.5 Data Augmentation

Data augmentation is a method used to artificially increase the size of a dataset by creating new training examples through various transformations or modifications applied to the original data. In this study, Independent Component Analysis (ICA) was utilized as an augmentation technique using MNE-Python library.

ICA is a signal processing technique that decomposes multichannel EEG signals into their constituent components, which represent distinct brain activities or sources. In this study, ICA was employed for data augmentation to accelerate the training process of the CNN model.

Second, ICA can introduce variability and diversity into augmented data by creating new combinations of independent components that are not present in the original set. This increases diversity within training examples and may improve generalization ability of a trained CNN model.

Furthermore, ICA is a well-established technique in EEG research for artifact removal and source localization, and its application to data augmentation has been reported as effective at improving machine learning models' performance on EEG analysis tasks (Kang et al., 2022, pp. 8-10). Therefore, ICA was chosen as an appropriate data augmentation method in this study in line with its objectives and hypotheses.

2.6 Data Preprocessing

There are many studies that have looked at the impact of different data processing and data acquisition techniques on EEG quality and subsequent analysis for BCI application (Pinegger et al., 2016, pp. 3-10). It is important to carefully consider the research objectives, experimental design and the characteristics of EEG signals to be used in selecting the best data acquisition and processing methods.

EEG data should be preprocessed to reduce noise, artifacts and improve quality before further analysis. The most common preprocessing methods include filtering and artifact removal. Filtering is used for removing noise from EEG signals. This includes powerline interference, muscle artifacts and electrode drift. EEG data can have a number of filters. (Robbins et al., 2020, p. 9.)

According to Jiang et al. (2019, pp. 1-2), the removal of artifacts from EEG signals is a crucial step in data preprocessing. Artifact removal methods are used to remove non-brain-related signals like eye blinks, eye movements, or cardiac artifacts that can contaminate EEG data. To identify and remove EEG artifacts, these techniques include regression-based or threshold-based approaches.

Re-referencing, another common preprocessing step, involves changing the source electrode to reduce the impact on reference-dependent artifacts like Common Average (CAR), or linked mastoids, which are fused reference electrodes into one connector plugged into a reference input. Reference electrodes provides a stable baseline against which the electrical activity of the brain is measured and are typically placed in a neutral or non-active area on the body to minimize the influence of electrical activity unrelated to the brain. The choice of a reference electrode can impact the interpretation of EEG signal and the performance in subsequent analysis, such features extraction and machine learning classification (Trujillo et al., 2017, p. 2).

Segmentation in EEG analysis refers to the practice of dividing an uninterrupted EEG signal into smaller segments or epochs for easier analysis of specific features or patterns within it, such as Event-Related Potentials (ERPs) or brain rhythms. Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT) are two signal processing techniques that can help with EEG signal segmentation. DWT divides signals into their various frequency components or scales while DFT breaks it into its constituent sinusoidal components at various frequencies (Procházka et al., 2010, p. 317).

Time-Frequency Analysis is a technique for examining how the spectral characteristics of a signal change over time. According to Morales & Bowers (2022, p. 4), time-frequency analysis methods can be used to analyze developmental EEG data. It provides insight into how different frequency components develop over time, making it especially helpful when studying dynamic signals such as EEG data.

Feature extraction is an essential element of EEG data analysis. This process converts raw EEG data into meaningful features that can be utilized further for modeling or other analyses (Azlan & Low, 2015, p. 801). Feature selection is the process of selecting a subset of relevant features from an extensive set. This reduces data dimensionality and enhances efficiency and interpretability during subsequent analysis (Lan, 2011, pp. 42-43).

Electrode pairs are selected to extract features from specific pairs of electrodes for training a neural network. Each electrode pair represents two electrodes from which data will be extracted for feature extraction (Lun et al., 2020, p. 3).

Preprocessing and data acquisition are key steps in EEG data analysis. This can greatly impact the accuracy and reliability, as well as the reliability, of the results. It is worth looking through the literature to find out the best practices and the latest techniques in data acquisition for EEG-based BCI research.

2.7 Machine Learning for EEG Signal Classification

Machine Learning, a subfield of artificial intelligence that allows computers to learn from data without explicitly programming it, has gained popularity due to its ability to quickly analyze large volumes of information and make accurate predictions. Machine learning has shown great potential in EEG signal classification (Abdeltawab & Ahmad, 2020, p. 200).

This literature review highlights machine learning techniques utilized for EEG signal classification when used to control a robot arm using motor imagery tasks. The primary objective was to develop a machine learning model capable of accurately predicting different motor imagery tasks using EEG signals and thus be used to control a robot arm.

2.7.1 CNN for EEG Signal Classification

Deep learning has emerged as a powerful technique for EEG signal analysis and has shown great potential in various applications in electroencephalography. This section of the literature review also presents current research developments in the application of deep learning techniques for EEG signal analysis Deep learning models, such as CNNs have been successfully applied to EEG data for event recognition, segmentation, feature extraction, etc. (Mao et al., 2020, p. 5.) These models can learn patterns and stability from raw EEG signal. In addition, CNNs has been used for brain-computer interface (BCI) applications, where it enables decoding motor imagery, mental state, or intention directly from EEG signals. CNN models have shown improved performance in real-time EEG-based control external devices. (Lawhern et al., 2018, pp. 1-2, 4-22.) Despite the successes, challenges remain in applying deep learning to EEG signal analysis. Apart from the limited availability of large labeled EEG datasets for training deep learning models, which presents challenges in model performance and overall, the interpretation of deep learning models in EEG in research are areas of active research (Cui et al., 2022).

2.7.2 Transfer Learning

Transfer learning is a machine-learning technique that uses knowledge from one task to improve performance in another task. Because it can reduce the limitations of data and require less training (Zhao, 2017, p. 6), transfer learning has been gaining significant attention in recent times. Transfer learning has been successfully applied in various fields, including computer vision, natural language processing, and speech recognition, and has also shown promise in the domain of BCIs and MI classification using EEG (Parvan et al., 2019, pp. 1827-1828).

Transfer learning in EEG based MI classification refers to the use of knowledge from one set or domain to improve classification performance on another set or domain. This can be done in a variety of ways including model adaptation, feature extraction, and domain adapt.

One approach to transfer learning for MI classification involves feature extraction. This is where features taken from one MI dataset can be used to extract relevant features in another MI dataset (Jha et al., 2022, p. 1). Features can be extracted, for example, from MI data from healthy individuals to extract features from MI records from people with motor impairments. This method improves classification accuracy as the model can capture relevant information from the source domain, and then transfer it to its target domain.

Model adaptation is another approach to transfer learning. This involves adapting a model that has been trained in the source domain to the target domain. Fine-tuning is where the pre-trained model from the source domain is further trained with the target domain data. Fine-tuning allows the model's performance to be improved and adapt to the characteristics of the target domain (Zhang et al., 2021, p. 2).

Domain adaptation is another method of transfer learning. The model is trained on a source site and then applied to a target site without explicitly using the target domain data (Ju et al., 2020, p. 3040). This is useful in situations where it is impossible or not practical to collect labeled information from the target domain. Domain adaptation techniques include domain adversarial learning, which trains the model to minimize differences between the source domain representations and the target domain representations. Also, domain-specific normalization is where data from both the target and source domains are normalized to one common distribution.

It has been proven that transfer learning can improve the classification performance for EEG-based MI-BCI’s. Transfer learning has been used to transfer knowledge between different MI tasks and across different populations (Wu et al., 2022, pp. 235 - 238) such as those with motor impairments and healthy people.

Transfer learning for EEG-based MI-BCI’s presents challenges. These include the domain shifting between source and target domains, the optimal selection of transfer learning methods, and the possibility of negative transfer where the knowledge of the source domain might not be of benefit to the target domain (Zhang et al., 2023, p. 1). It is important to carefully evaluate the suitability of each transfer learning approach and how they can be applied to specific MI-BCI applications.

Transfer learning, which leverages knowledge from one MI domain or MI dataset to improve classification performance on another MI dataset/domain (Zhang et al., 2021, p. 7), is a promising way to improve the performance of EEG based MI-BCI’s. Although it has been proven to work in many scenarios and applications, there are still limitations and challenges. To fully harness the potential of EEG-based MI-BCI’s using transfer learning, further research and development is needed.

2.8 Robot Control using EEG

Robot control using EEG signals involves using machine learning algorithms to classify EEG signals recorded during motor imagery or other cognitive tasks and use those classifications to direct the movement of a robotic device. Utilizing EEG signals as a method for robot control have several advantages such as non-invasiveness, low cost, and the capacity to operate devices without physical movements required.

When it comes to robot control, there are two primary approaches: online and offline methods. Online methods involve real-time execution of robotic devices using EEG signals; on the other hand, offline methods use previously recorded EEG data in order to train machine learning models for robot action.

One common approach to EEG-based robot control utilizes machine learning algorithms such as support vector machines (SVMs) (Hortal et al., 2015) or CNNs that classify EEG signals into distinct classes associated with different movements or commands (Jeong et al., 2020, p. 1227). The output of this machine learning algorithm then guides the movement of a robotic device such as an arm.

EEG-based robot control has shown promise for a range of applications, such as neurorehabilitation, assistive technologies (Korovesis et al., 2019, p. 2) and industrial automation (Douibi et al., 2021, pp. 3-6). However, several challenges must be addressed to enhance its reliability and effectiveness: creating more robust machine learning algorithms; optimizing EEG signal acquisition/processing techniques; and integrating EEG-based robot control with other robotic technologies.

3 METHODOLOGY

3.1 Experimental Design

The experimental design involved multiple steps, such as EEG headset and electrode design and fabrication, data acquisition and preprocessing, machine learning model training and evaluation, and robot control.

The EEG headset and electrode design were created from a 3D-scanned and printed model of the participant's head, with electrode positions marked on it. To improve conductivity and reduce discomfort during use, different materials and coatings were utilized on both components.

Data acquisition was carried out with an OpenBCI board and custom software application that enabled data acquisition and visualization. To preprocess the EEG data, various techniques such as filtering and feature extraction were applied.

Machine learning models were trained on preprocessed EEG data, with several different CNN architectures tested to find the most accurate one. Their accuracy, precision, and recall were then assessed against a test dataset.

Finally, robot control was implemented using the ROS MoveIt package, which allowed for the prediction of Motor Imagery based on EEG signals and movement of a robot arm in response to those predictions.

Overall, this experimental design enabled successful acquisition, processing and analysis of EEG signals for robot control with custom-designed EEG headset and electrodes, machine learning models and robot control software.

3.2 EEG Headset Design and Fabrication

The design of the EEG headset was done to enhance user comfort during measurements while maintaining a good EEG signal (Zhu et al., 2021, p. 4). The headset design was intended to make EEG measurements less tedious for the user by being simple to wear and remove.

A 3D scanned model of the head was used to design the headset, marking where electrodes would go. The design process involved 3D scanning a human head to create a 3D model of the head, marking electrode positions on this model. The design was made for three electrodes placed on the scalp in C3, Cz and C4 positions.

The EEG headset and electrode system were created using computer-aided design (CAD) software. The EEG headset was constructed using 3D printing technology where various materials for printing were considered aiming for lightweight, robust and flexible material. It featured a 3D printed frame that fit over the head, supporting electrodes.

This headset and electrode system was designed with both hard and soft, flexible electrodes with various coatings for good electrical conductivity. Resistance, impedance and signal quality measurements of electrode materials and coatings were conducted to identify which materials and coatings work best in EEG signal acquisition.

3.2.1 3D Scanning and Printing of head

The initial step in designing and fabricating an EEG headset was creating a 3D model of a head using 3D scanning technology to capture its shape and size. Multiple scanners and software such Revopoint Mini scanner with Revopoint Studio software and Kinect 3D scanner with Skanect software were tested until finding the ideal scanner. The following figure (Figure 1) shows the set up used for 3D scanning of the head using the device and software finally used, which was Creality CR-Scan Lizard 3D scanner and CR-Studio software.

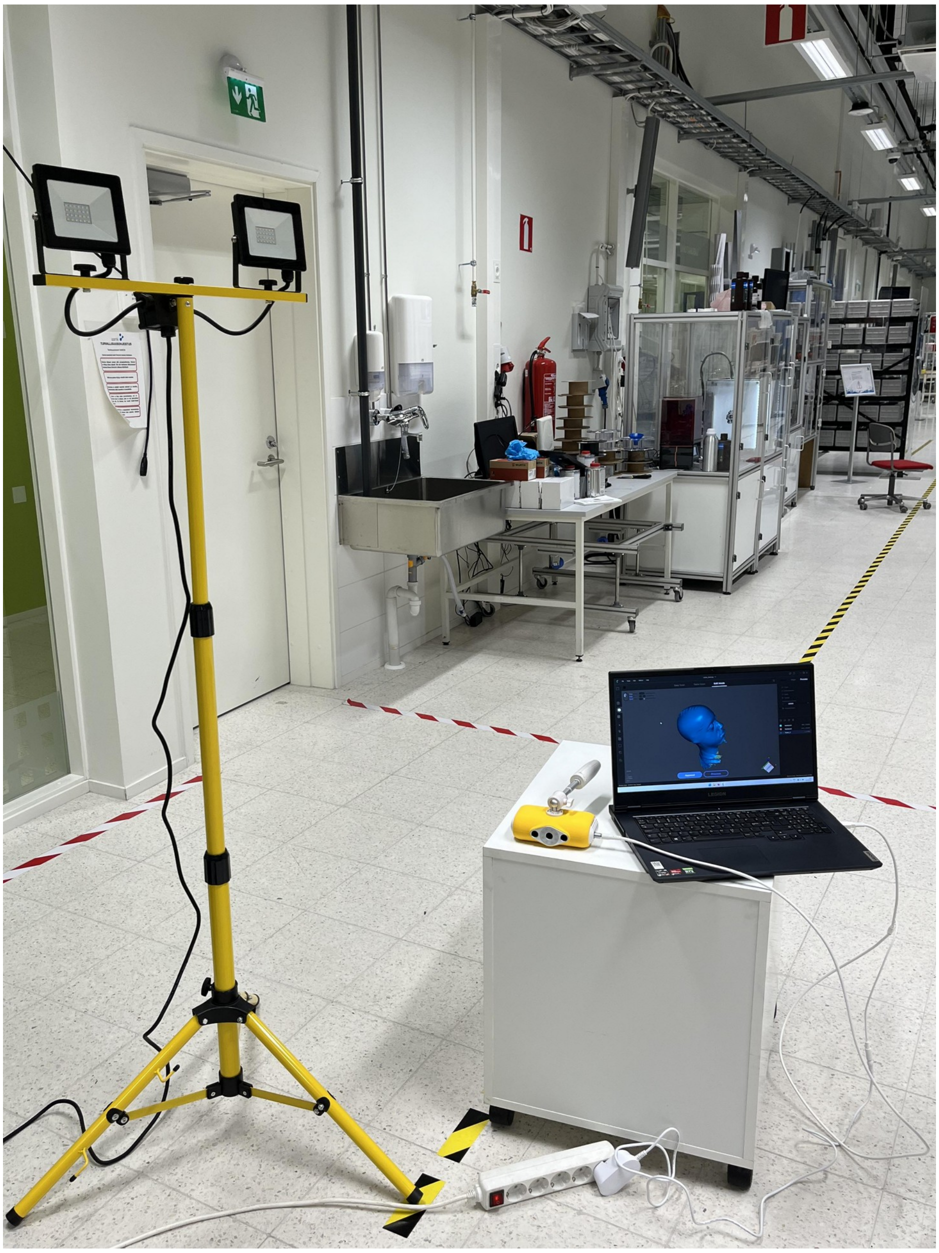


Figure 1. Set up for 3D scanner with Creality CR-Scan Lizard 3D scanner and CR-Studio software

Following scanning, an.OBJ file was obtained and converted to .STL format for further processing. MeshMixer was chosen for smoothing the head model as it proved more user-friendly compared to Blender; and then RobustSmooth technique was employed to achieve desired levels of smoothness. Figure 2 depicts the resulting 3D modeled head after the smoothing technique was applied using MeshMixer software.

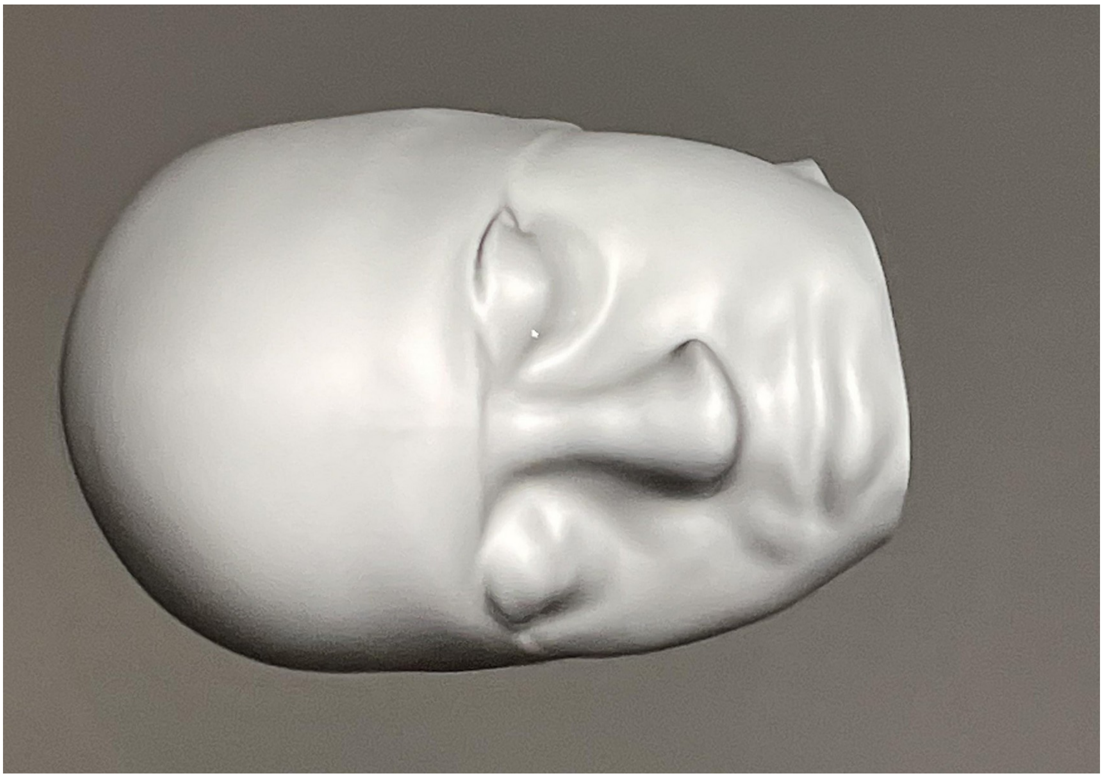


Figure 2. 3D modeled head after smoothing technique was applied

BambuStudio handled the preparation for printing, also known as slicing. As shown in Figure 3, Bambu Lab’s X1 printer was chosen due to the limited build volume on the Prusa i3 MK3s printer. PLA, from Azure Film, was used for this printing process at +215 degrees Celsius with a nozzle diameter set to 0.4 mm and layer thickness set at 0.2 mm during this phase of printing.

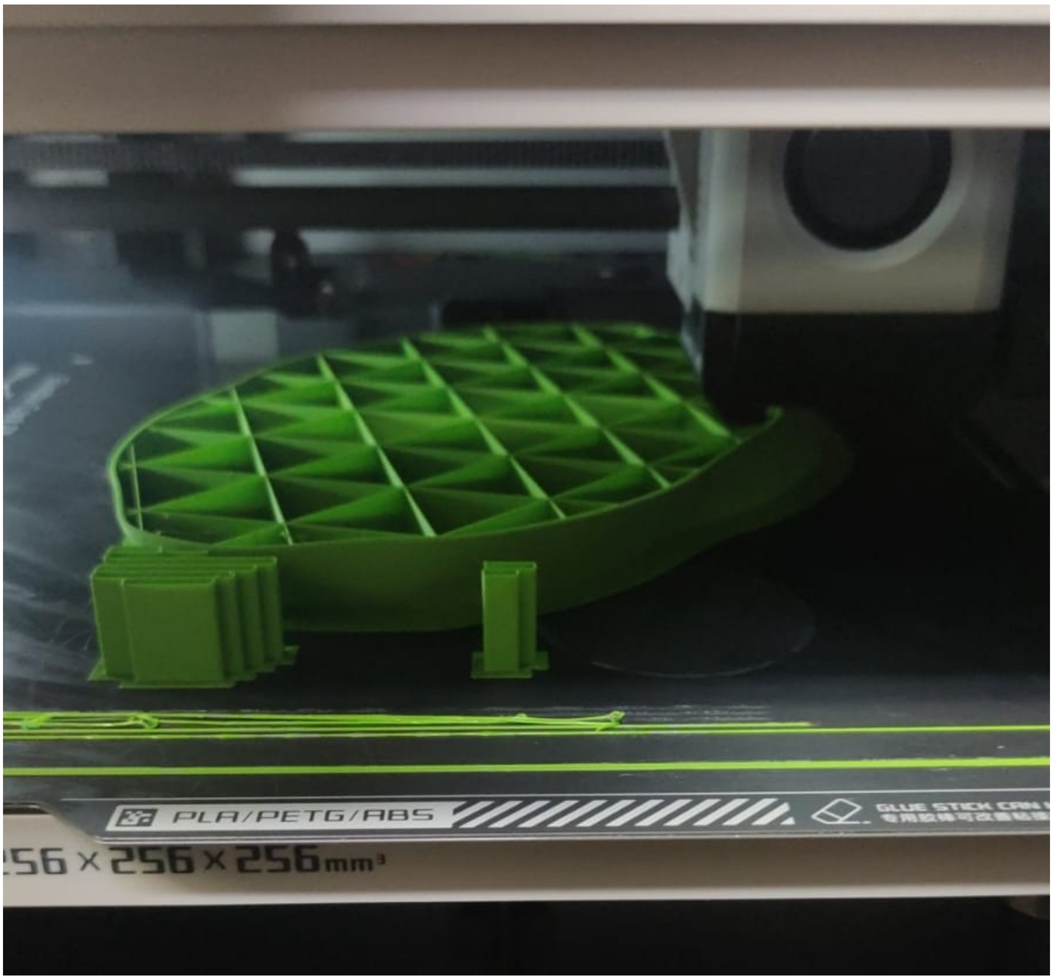


Figure 3. 3D modeled head being printed

3.2.2 EEG Channels Positioning and Rationale for Selection

After the 3D scanned head was successfully printed, the EEG channels were carefully selected and placed at locations C3, Cz and C4 on its surface based on their relevance for detecting MI within the brain (Lin & Shih, 2018, pp. 18-26). Also, locations A1 and A2 were used as reference points. Only three electrodes were used with the objective of reducing the amount of channels used, which would help to simplify the headset design and enhance comfortability.

As shown in Figure 4, the C3, Cz, and C4 positions correspond to the international 10-20 system, which is a widely used standard for electrode placement in EEG recordings. These positions are specifically associated with the motor cortex (Yahya et al., 2019, p. 2) which controls motor function and movement planning in the brain. Placing channels at these positions enable identification of MI-related electrical activity during EEG recordings.

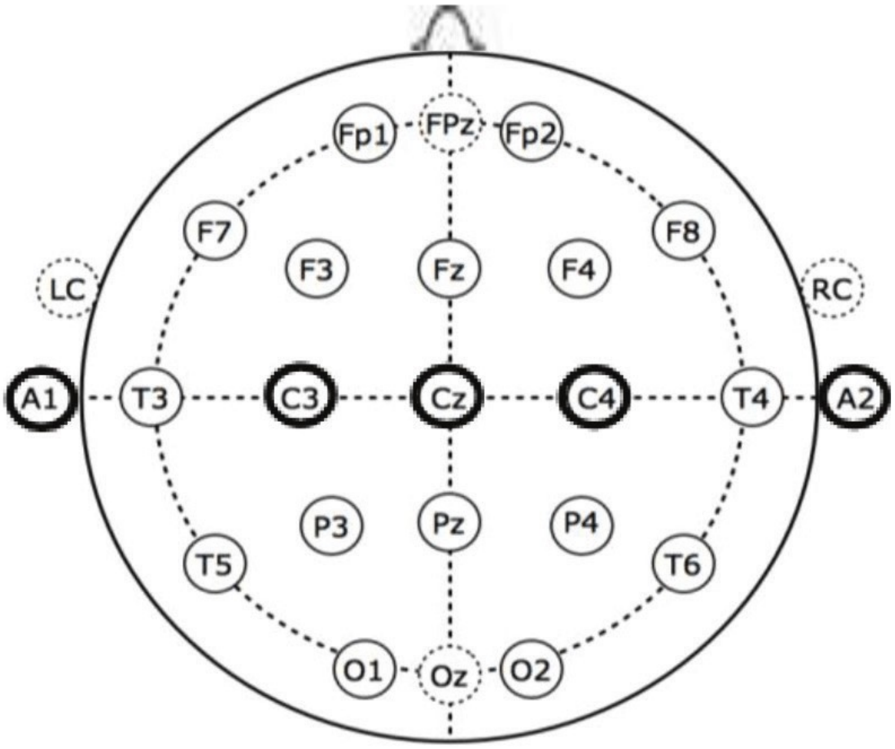


Figure 4. Locations C3, C4, and Cz are used in the 10-20 system (Lin et al., 2018)

For optimal EEG channel placement on the physical head, measurements were taken to calculate precise positions for C3, Cz, and C4 according to the individual's head size and anatomy. Once calculated, these positions were marked on a 3D model of the head to ensure accurate channel alignment with motor cortex for detection of MI-related brain activity.

3.2.3 3D Modeling and Printing of headset

The design of the EEG headset required taking into account both the size and shape of a 3D modeled head as well as the desired EEG channel locations previously selected. Inspired by the design of headphones, the headset was modeled using Fusion360 software. It was designed to fit the head perfectly as it can be observed in Figure 5, with a thin yet robust and slightly flexible design. Spaces were incorporated in the channel positions to allow for the electrodes to be inserted and tightly attached to the headset.

Also, included on each side of the headset were two cylindrical structures to hold rubber bands for easier attachment and to provide slight pressure so electrodes could make good contact with the scalp.

The Initial printing of the headset was done using Armadillo material, which is slightly more flexible but less resistant to breakage compared to PLA. As a result, PLA was chosen as the material for final print to ensure durability and reliability.

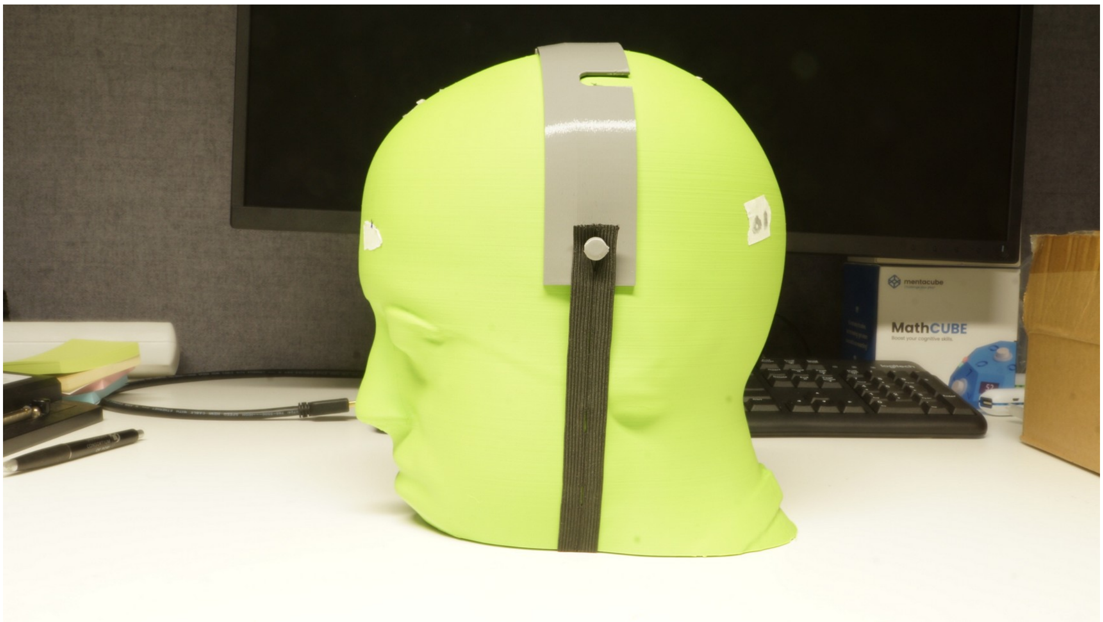


Figure 5. 3D printed headset

A Prusa i3 MK3s printer was used to print the headset with a temperature setting of +215 degrees Celsius and using a nozzle diameter of 0.4 mm with layer thickness set at 0.2 mm for the printing process. It was taken great care to ensure high-quality printing that would provide precise and accurate fitment of the headset on our tester's head.

3.3 3D Modeling and Printing of Electrodes

The 3D modeling and printing of electrodes process involved several key steps to ensure the optimal design, material selection and coating selection for reliable and accurate EEG measurements (Krachunov & Casson, 2016, pp. 2-8). The process started with the creation of electrode designs using Fusion360 (CAD) software, taking into consideration the requirements of the EEG headset and the desired fit on the scalp. Materials were carefully chosen based on their conductivity, flexibility, durability, and biocompatibility. Among the materials that were used are Conductive Filaflex, Eryone TPU-Transparent, ProtoPasta Conductive Filament, Anycubic Grey, Siraya Tech Blu and Resin F80. Coatings such as Electrically Conductive Paint and Bare Conductive Electric Paint were selected to ensure optimal electrical contact. Resistance and impedance testing were then conducted to assess their electrical performance. The following sections provide detailed information regarding electrodes designs, material selection, coating selection, resistance testing, impedance testing and signal quality testing, emphasizing on their rationale and the importance within 3D modeling and printing of electrodes.

3.3.1 Electrode Designs

The design of the electrodes is a crucial aspect of 3D modeling and printing process, it directly impacts the performance and comfort of EEG measurements. Fusion360 software was used to create various designs of electrodes to ensure optimal fit and electrical contact on the scalp within the requirements of the EEG headset.

Each electrode design was meticulously created to meet the specific requirements of the EEG headset and to guarantee maximum comfort during EEG measurements, taking into consideration the materials' properties and printing capabilities.

Four different electrodes designs were developed:

1. Headpin Flat: This electrode design was inspired by the Enobio flat electrode (Figure 6) and intended for use on hairless parts of the head such as the forehead. However, it was not used in the final prototype as the EEG channel positions of interest were located in hairy areas of the head, specifically C3, Cz, and C4.

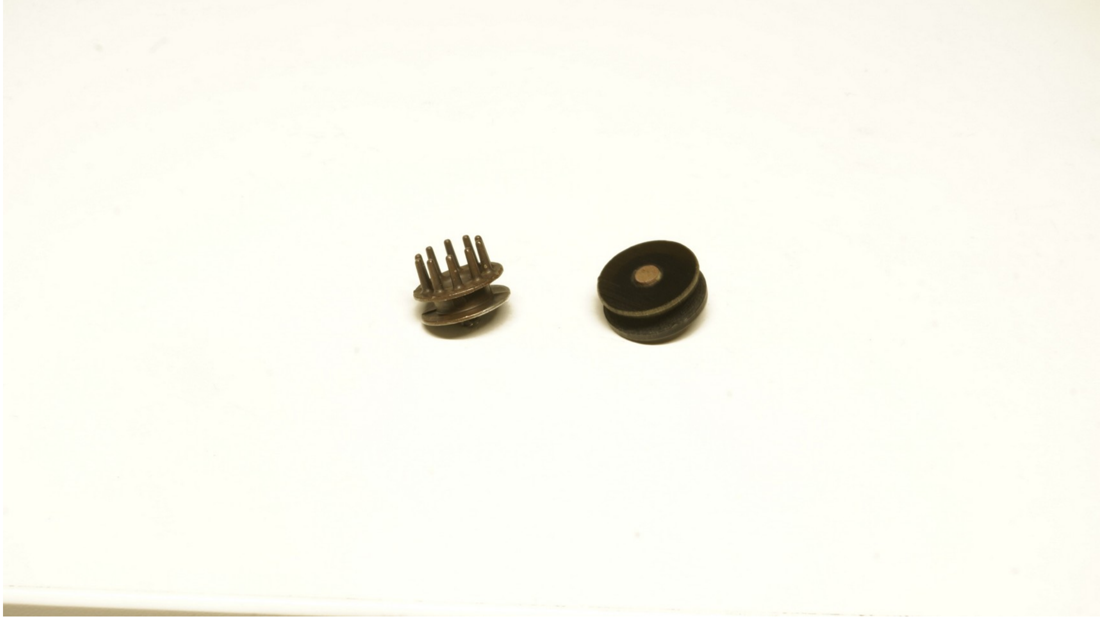


Figure 6. Original Enobio Multiplestuds (left) and flat (right) electrodes

2. Headpin MultipleStuds: This electrode design was inspired by the Enobio electrode (Figure 6) for hairy areas of the head, with two additional spikes. It was originally designed to work on non-flexible materials, but it was also printed using flexible materials for testing purposes.
3. Headpin Spiderman: This design was specially created for flexible materials. It features the shape of a spider with slightly outward-facing spikes. Additionally, shorter spikes were positioned inside the electrode to make

contact with the head when the electrode bends, resembling the legs of a spider.

4. Headpin Venom: This design is similar to the Spiderman Headpin, with the exception that the spikes at the center of the electrode have the same length as the outer spikes (Figure 7). It was specially designed to be printed with non-flexible materials, although it was also printed with flexible materials for testing purposes.

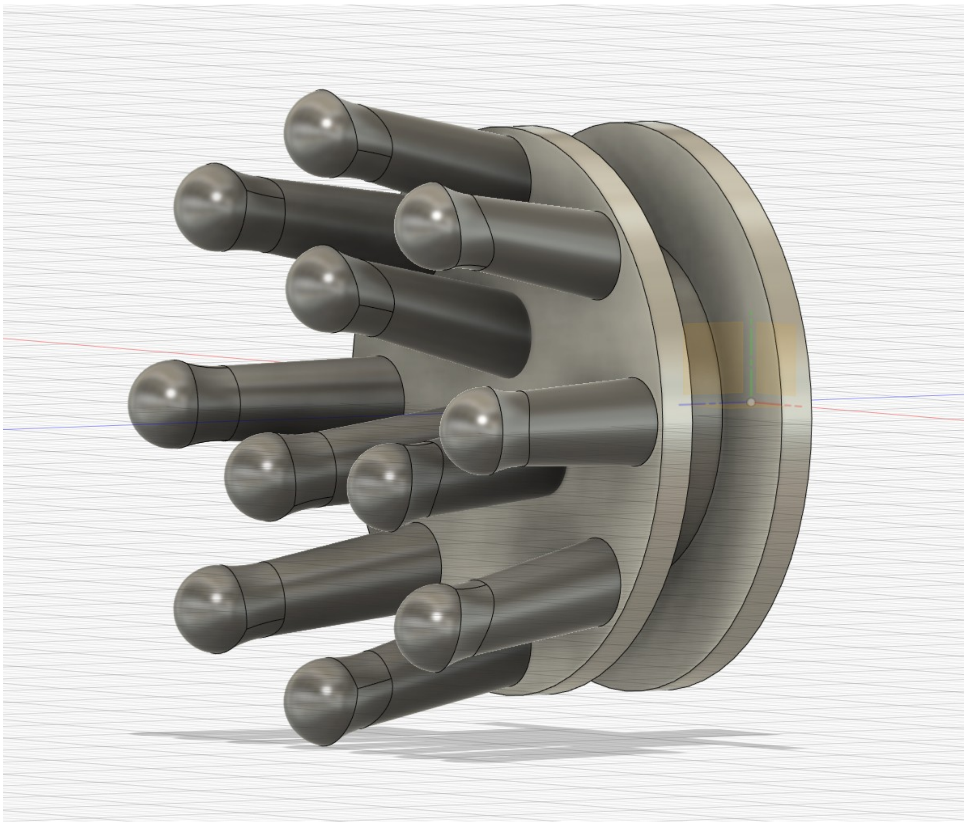


Figure 7. Headpin Venom modeled electrode

3.3.2 Material Selection and Rationale for Selection

The selection of materials for 3D printing of the electrodes was a critical factor in ensuring their conductivity, flexibility, durability, and biocompatibility. Several materials were considered including both conductive and non-conductive options. For the non-conductive materials, different conductive paints were applied as coating using a brush to enhance their conductivity. This approach allowed for the utilization of multiple materials while still meeting electrical conductivity requirements. The following materials were selected for their unique properties:

3.3.2.1 Conductive Filaflex

Conductive Filaflex, a flexible filament infused with conductive particles, was chosen due to its superior electrical conductivity and flexibility. This material was chosen for 3D printing of electrodes as it conformed well with scalp contours while offering reliable electrical contact during EEG measurements (Velcescu et al., 2019, pp. 2-4).

Recreus manufactured this Flexible Conductive filament with a diameter of 1.75 mm for printing at temperatures up to 255 degrees Celsius using a nozzle diameter of 0.6 mm, although initial tests with 0.4 mm proved unreliable due to frequent clogging issues; ultimately requiring us to change to larger diameter nozzle. Layer thickness was set at 0.3 mm on our Prusa i3 MK3S printer. The resulting electrodes can be seen in Figure 8 where Headpin Spiderman, MultipleStuds, and flat electrodes were printed with Filaflex conductive material.

Conductive Filaflex filament can be fragile and susceptible to breakage when left uncoated, leading to frequent breakages when in use. To address this challenge and enhance conductivity, Conductive Filaflex was coated with various conductive paints for increased durability and resistance; coating was also performed for testing purposes.

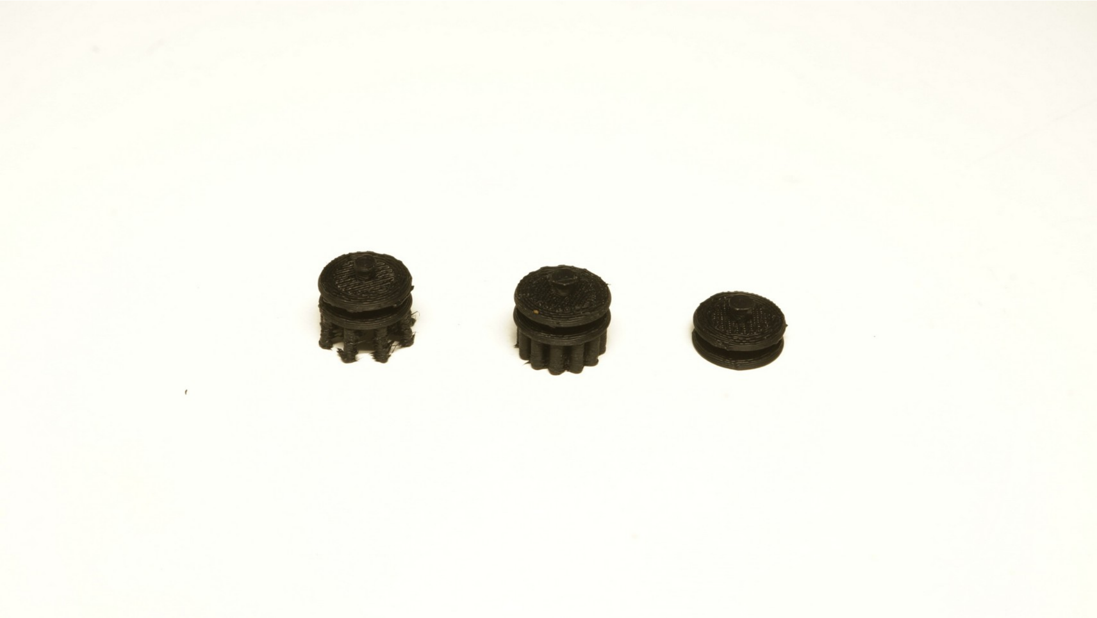


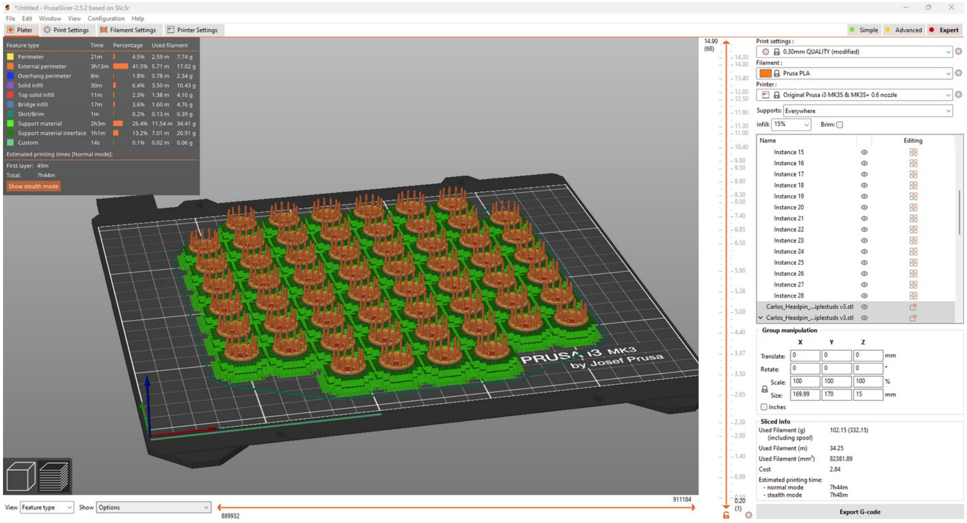
Figure 8. Headpin Spiderman (left), MultipleStuds (center) and flat (right) electrodes printed with FilaFlex conductive material

3.3.2.2 Eryone TPU

Eryone TPU filament was selected due to its flexibility and durability, making it suitable for conforming to the contours of scalp during EEG measurements. Furthermore, 3D printing makes this choice even more suitable.

Eryone material required printing parameters that ranged between 230-240 degrees Celsius, with a nozzle diameter of 0.4 mm and layer thickness of 0.2 mm for reliable and consistent results. A Prusa i3 MK3S printer was employed for this process and results were consistently achieved. Figure 9 shows the preparation of Headpin MultipleStuds filament electrodes in PrusaSlicer software for 3D printing (slicing).

Eryone TPU itself is not electrically conductive; various conductive paints were employed as coatings on it in order to maximize flexibility while simultaneously meeting the electrical conductivity requirements for electrodes.



Feature type	Time	Percentage	Used filament	
🟨 Perimeter	21m	4.5%	2.59 m	7.74 g
🟧 External perimeter	3h13m	41.5%	5.71 m	17.02 g
🟦 Overhang perimeter	8m	1.8%	0.78 m	2.34 g
🟪 Solid infill	30m	6.4%	3.50 m	10.43 g
🟥 Top solid infill	11m	2.3%	1.38 m	4.10 g
🟦 Bridge infill	17m	3.6%	1.60 m	4.76 g
🟩 Skirt/Brim	1m	0.2%	0.13 m	0.39 g
🟩 Support material	2h3m	26.4%	11.54 m	34.41 g
🟩 Support material interface	1h1m	13.2%	7.01 m	20.91 g
🟩 Custom	14s	0.1%	0.02 m	0.06 g

Estimated printing times [Normal mode]:

First layer: 49m
Total: 7h44m

Print settings : 0.30mm QUALITY (modified)
Filament : Prusa PLA
Printer : Original Prusa i3 MK3S & MK3S+ 0.6 nozzle
Supports: Everywhere
Infill: 15% Brim: □

Name		Editing
Instance 15	👁	
Instance 16	👁	
Instance 17	👁	
Instance 18	👁	
Instance 19	👁	
Instance 20	👁	
Instance 21	👁	
Instance 22	👁	
Instance 23	👁	
Instance 24	👁	
Instance 25	👁	
Instance 26	👁	
Instance 27	👁	
Instance 28	👁	
Carlos_Headpin_...iplestuds v3.stl	👁	
Carlos_Headpin_...iplestuds v3.stl	👁	

Group manipulation

	X	Y	Z	
Translate:	0	0	0	mm
Rotate:	0	0	0	°
Scale:	100	100	100	%
Size:	169.99	170	15	mm

□ Inches

Sliced Info

Used Filament (g) (including spool)	102.15 (332.15)
Used Filament (m)	34.25
Used Filament (mm³)	82381.89
Cost	2.84
Estimated printing time:	
- normal mode	7h44m
- stealth mode	7h48m

Figure 9. Headpin MultipleStuds filament electrodes being prepared using PrusaSlicer software for slicing

3.3.2.3 ProtoPasta Conductive Filament

ProtoPasta Conductive Filament was chosen for its excellent electrical conductivity. Produced by Protopasta, this non-flexible but highly conductive filament offers reliable and consistent electrical performance that makes it suitable for making accurate EEG measurements.

Printing parameters for ProtoPasta Conductive Filament included 240 degrees Celsius, with a nozzle diameter of 0.4 mm and layer thickness of 0.2 mm. This material had a larger diameter of 2.85 mm and was printed using an Ultimaker 3 printer; however, one challenge with this material is its difficult printing nature requiring careful handling and adjustments to achieve optimal results.

3.3.2.4 Anycubic Grey

Anycubic Grey, a durable and non-flexible resin from Anycubic, was chosen due to its robustness and potential for achieving long-lasting electrodes. This material was 3D printed with an Elegoo Mars 3 printer using an exposure time of 2.5 seconds and layer thickness of 0.05 mm. Figure 10 shows the preparation of a Headpin MultipleStuds resin electrode using Anycubic software Photon Workshop.

Anycubic Grey may not be intrinsically conductive; therefore, various conductive paints were used to optimize its conductivity and meet the desired electrical conductivity requirements for the electrodes.

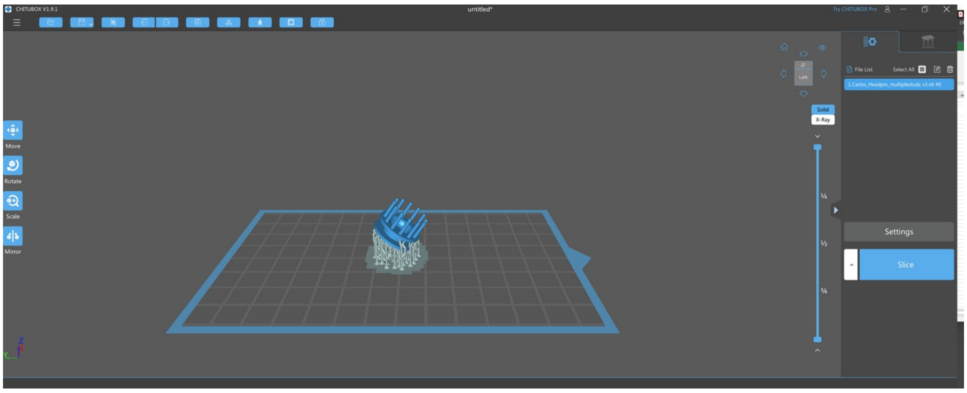


Figure 10. Headpin MultipleStuds resin electrode being prepared for 3D printing

3.3.2.5 Siraya Tech Blu

Siraya Tech Blu resin was selected due to its outstanding mechanical strength and resistance to wear and tear, making it suitable for producing electrodes that require exceptional durability, such as electrodes used in long-term EEG monitoring, where extended usage and potential physical stresses may be encountered.

Siraya Tech Blu, manufactured by Siraya, was 3D printed using an Elegoo Mars 3 printer and set its parameters with an exposure time of 4 seconds and layer thickness of 0.05 mm. One challenge faced during printing involved finding an optimal exposure time, which may require multiple iterative adjustments, as well as issues related to temperature regulation in their printing shop. The output of the printing can be observed in Figure 11 where Headpin Venom, MultipleStuds, and flat electrodes, were printed using this resin. As this material was not inherently conductive, different conductive paints had to be applied in order to provide electrical conductivity.

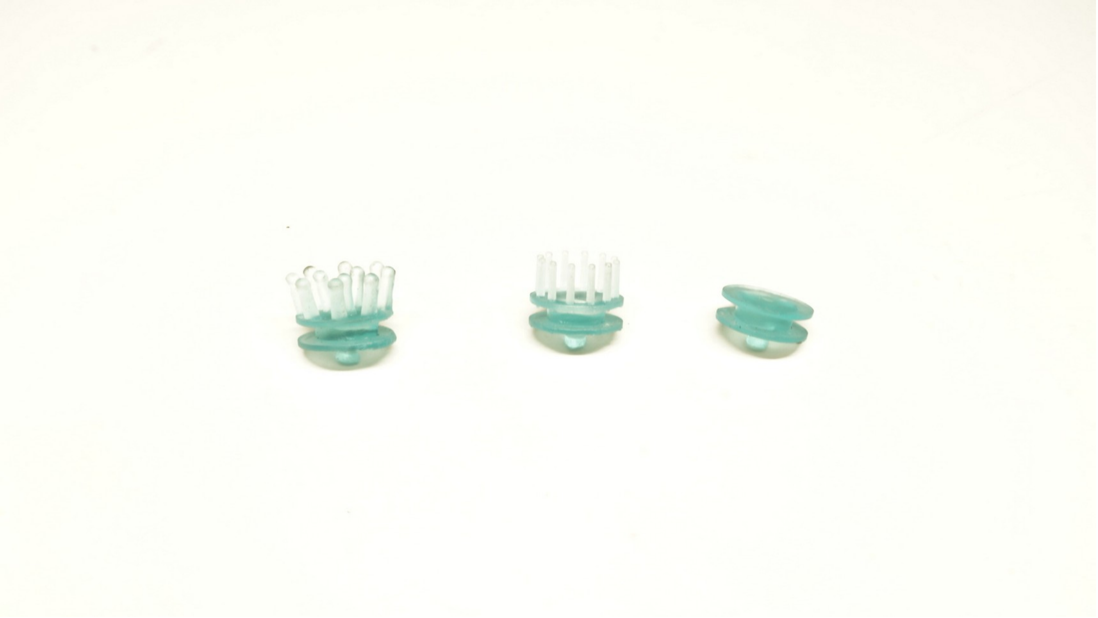


Figure 11. Headpin Venom (left), MultipleStuds (center) and flat (right) electrodes printed with Siraya Tech Blu material

3.3.2.6 Resione F80

Resin F80 from Resione, which is a biocompatible and flexible resin, was chosen due to its biocompatibility and flexibility, making it an ideal material for electrodes that require direct skin contact during EEG measurements. Resione's manufactured product ensures reduced risk of irritation or discomfort for users during extended EEG measurements, offering them a safe and pleasant experience.

Resin F80 was printed using an Elegoo Mars 3 printer with an exposure time of 7.5 seconds and layer thickness of 0.1 mm, but challenges in finding an adequate exposure time and temperature challenges in printing shops caused issues similar to Siraya Tech Blu material. For testing purposes, and as it can be seen in Figure 12, only Headpin Venom and flat electrode designs were printed with this material. Resin F80 itself is nonconductive, so to meet EEG measurement specifications it requires additional coating with electrically conducting paints in order to achieve electrical conductivity.

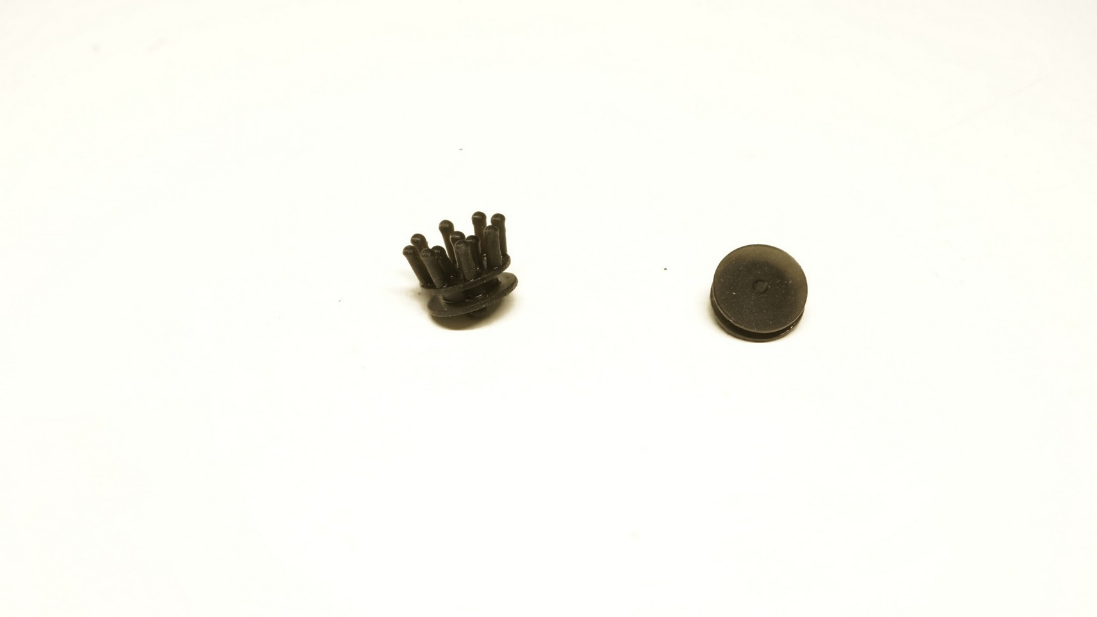


Figure 12. Headpin Venom (left) and flat (right) electrodes printed with Resione F80 resin material

3.3.3 Coatings Selection

Coatings play an essential role in providing good skin contact by improving resistance and impedance values during EEG measurements (Fiedler et al., 2009, p. 419). Giving enormous importance of selecting appropriate coating options, a careful evaluation was conducted, considering various factors such as their compatibility with materials chosen and their electrical conductivity properties. After thorough analysis, the following coating options were chosen:

3.3.3.1 Electrically Conductive Paint

Electrically conductive paint is composed of a thick mixture that dries quickly, making it perfect for coating EEG electrodes due to its electrically conductive properties. As it is water-based and free from metals, it is also safe for sensitive skin areas. Viscosity can be adjusted by diluting it with water. It was mixed with 100 drops of water using measuring tools (Figure 13) in order to create a fluid consistency suitable for coating purposes.

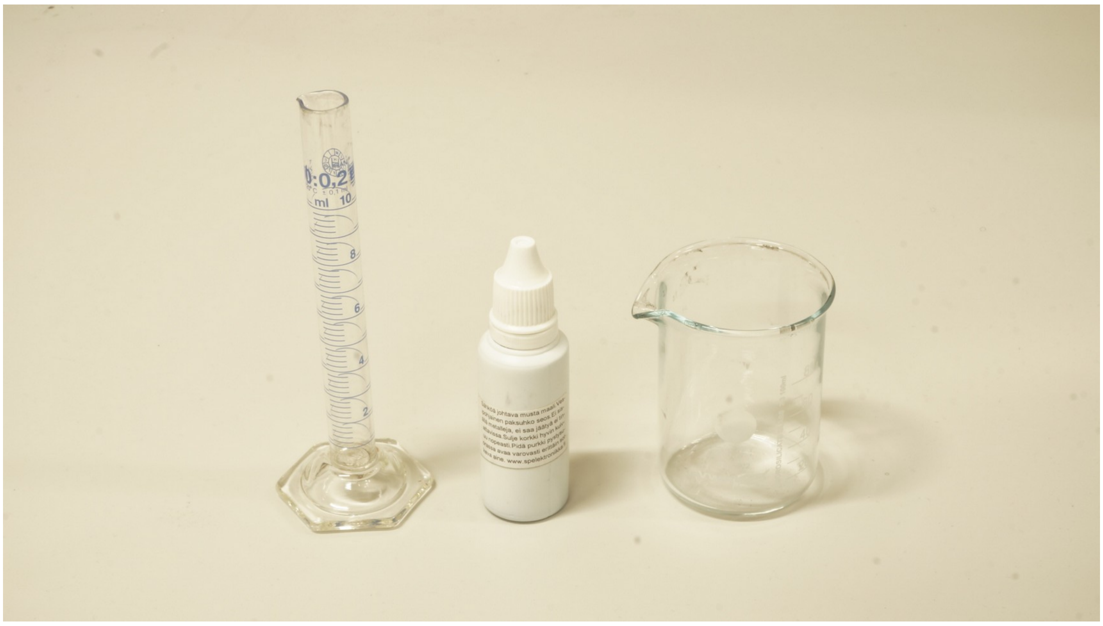


Figure 13. Electrically conductive paint and tools used for diluting it in 100 drops of water

Electrically conductive paint applied to EEG electrodes creates an effective layer that facilitates electrical signals being transferred from scalp to electrodes and allows accurate measurements during EEG recordings.

One advantage of this paint is its resistance to water exposure, making it suitable for applications where EEG electrodes may come into contact with moisture during measurements.

Overall, electrically conductive paint diluted with water can be used to coat EEG electrodes and form a conductive layer for measuring electrical signals from the scalp while offering excellent water resistance.

3.3.3.2 Bare Conductive Electric Paint

Bare Conductive Electric Paint (Figure 14) is water-based, non-toxic paint designed to dry at room temperature, making it safe and convenient to work with. Additionally, its strong adhesion properties make it suitable for EEG electrodes.

Bare Conductive Electric Paint stands out as an outstanding electrical conductor due to its highly effective conductivity, which makes it ideal for creating an EEG electrode conductive layer and efficiently transporting signals between scalp and electrodes during EEG measurements. With sheet resistance levels as low as 55Ω/sq at 50 µm film thickness, this paint provides reliable and effective electrical conduction.

Figure 14. Bare Conductive Electric Paint

Bare Conductive Electric Paint offers the combination of electrically conductive, water-based, non-toxic, and air-drying properties that makes it perfect for coating EEG electrodes. By creating an extremely effective conductive layer during EEG recordings, Bare Conductive Electric Paint facilitates measurement of scalp electrical signals for accurate EEG recordings. This makes it an ideal choice for researchers and practitioners in the field of neuroscience who require a reliable and convenient solution for coating EEG electrodes.

3.3.4 Resistance, Impedance and Signal Quality Testing

Resistance, impedance and signal quality tests were conducted to evaluate the performance of the 3D printed EEG headset and electrode system designed specifically for recording EEG signals. Each test consisted of measuring five electrodes of each material and coating and the results were averaged in order to obtain a reliable result.

Resistance can still be measured in the context of EEG signal quality because while impedance may be more relevant for evaluating the quality of EEG signals due to their alternating current nature, resistance provides valuable information about the direct current flow and can help identify issues such as poor electrode contact. For resistance testing, a multimeter was used in order to determine the conductivity of the electrodes printed with multiple materials and different coatings. Multimeter was set to resistance mode and the electrodes were tested using tweezers for avoiding hand shaking movements. Each tweeze was attached to each end of the electrode, one in the head of an electrode’s spike and the other one at the end where the electrode is attached to the cable.

To conduct impedance testing, the OpenBCI Cyton board with OpenBCI software was employed. Designed specifically for EEG measurements and equipped with impedance measurement capabilities, electrodes were connected to this board and placed making contact with the user's scalp while OpenBCI software measured their impedance using impedance measurements. Such measurements provide important insight into electrical characteristics of electrodes such as how effectively they conduct signals at various frequencies.

Utilizing the multimeter and OpenBCI Cyton board with OpenBCI software enabled comprehensive resistance and impedance testing of EEG electrodes. Impedance testing allowed for accurate and reliable measurements, providing valuable data about their performance and quality in relation to EEG recordings. Then, the results were compared with commercial EEG electrodes.

Measuring Signal Quality was assessed utilizing a subjective method. Electrodes of each material and coating were placed in the headset, which was wore by the user. The signal quality that each electrode produce was ranked from 1 to 5, being "1 Very Good Signal", "2 Good Signal", "3 Normal signal", 4" Bad Signal" and "5 Very Bad Signal". The ranking of signal quality was determined by two factors: artifacts and noise. These factors were observed when visualizing real-time EEG data while using each type of electrodes and coatings.

3.4 EEG Data Acquisition

Data acquisition for EEG data acquisition involved recording the electrical activity of the brain with an EEG headset and electrodes. Measurements were taken while participant performed motor imagery tasks, imagining movement of their limbs without actually performing it.

The OpenBCI Cython board was used to capture EEG signals from electrodes, and it was connected to a computer using different techniques such as Lab Streaming Layer (LSL), BrainFlow, and ROS. After comparing these different approaches, a Python code utilizing BrainFlow library for connecting with the board was selected as the most efficient and effective method for EEG signal acquisition.

Participant was instructed to perform motor imagery tasks while EEG measurements were being recorded. Once collected, data was preprocessed to extract useful features for classification by filtering signals to remove noise, segmenting them into appropriate time windows, and transforming it into a suitable format for machine learning analysis. No external participants were necessary for this study as the data acquisition was done through online public datasets and individual measurements. Therefore, as a sole participant in the study, no participation agreement or ethical document was needed.

3.4.1 Data collection Procedure

Desai's (2014, pp. 43-48) thesis provided valuable insights into the collection of EEG data. The methodology described in the thesis was particularly inspiring, and it served as a model for the data collection process.

Data was obtained using a Python code for data acquisition. These were the steps involved in data collection:

1. Headset: Enobio Neoprene headcap size M was utilized in order to realize EEG measurements while 3D printed headset was designed and printed. Enobio Neoprene headcap (Figure 15) has a total of 39 positions where electrodes can be placed, based on a subset of the international 10-10 EEG system.

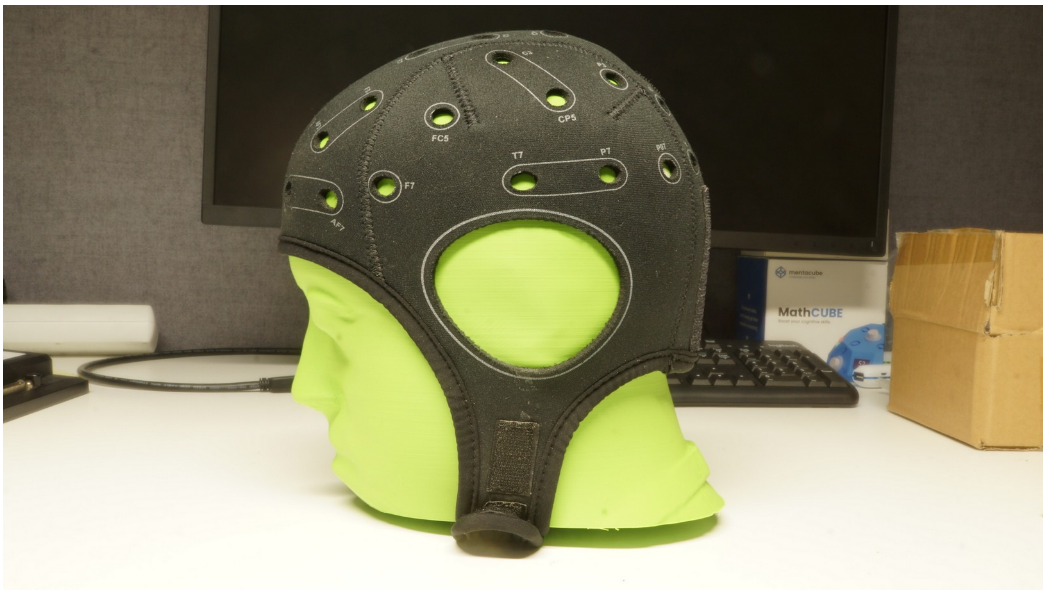


Figure 15. Enobio Neoprene headcap size M

2. Electrode placement: 3 Enobio electrodes were utilized while 3D printed electrodes were being printed, coated and tested. These electrodes were placed in positions C3, Cz and C4 plus two reference OpenBCI earclip electrodes (Figure 16), made of Silver-silver Chloride, which were placed on each earlobe (positions A1 and A2 in international 10-10 EEG system).

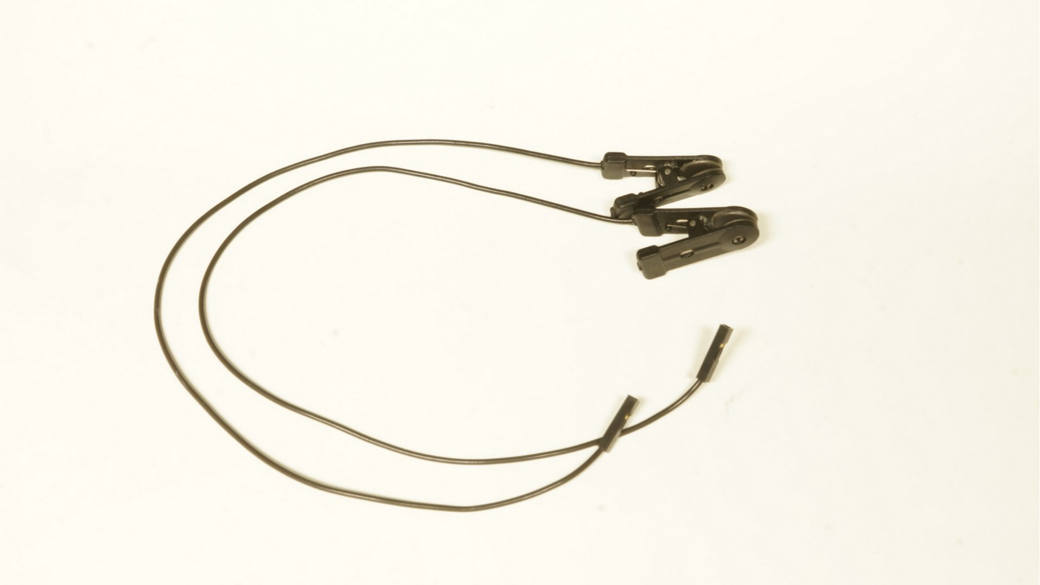


Figure 16. OpenBCI earclip reference electrodes

3. Board and Dongle: OpenBCI 8-channels Cyton board was used for data acquisition connecting to electrodes through OpenBCI EMG/ECG Snap electrode cables. Cyton dongle was utilized for connecting the Cyton board with computer through Bluetooth. The setup for data acquisition using OpenBCI equipment is shown in Figures 17 and 18, with Figure 17 displaying the OpenBCI Cyton board and dongle, while Figure 18 shows the OpenBCI EMG/ECG Snap electrode cables.

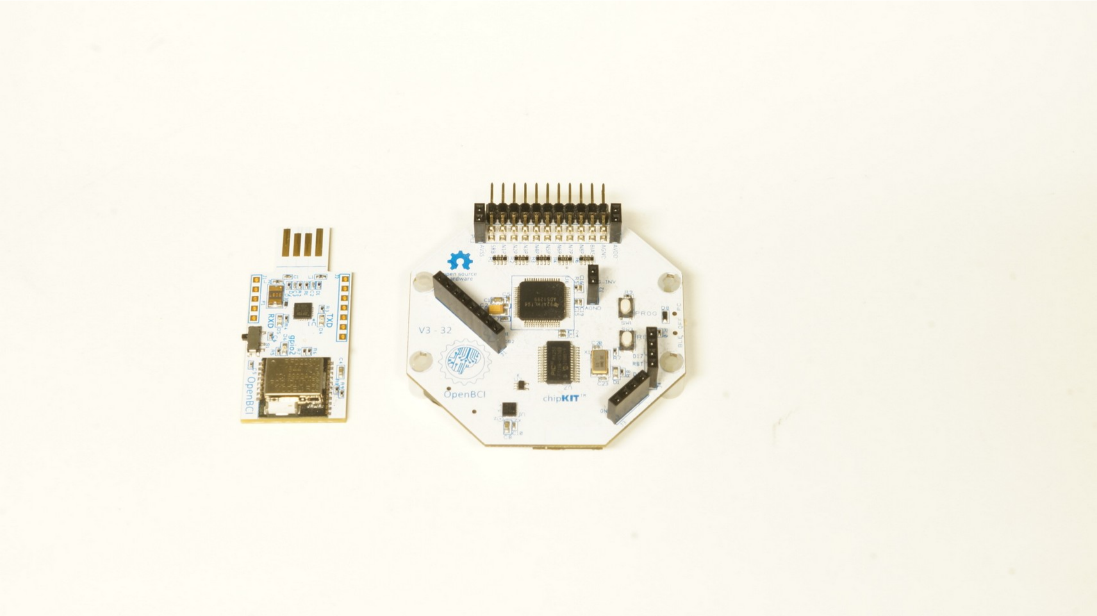


Figure 17. OpenBCI Cyton board and dongle

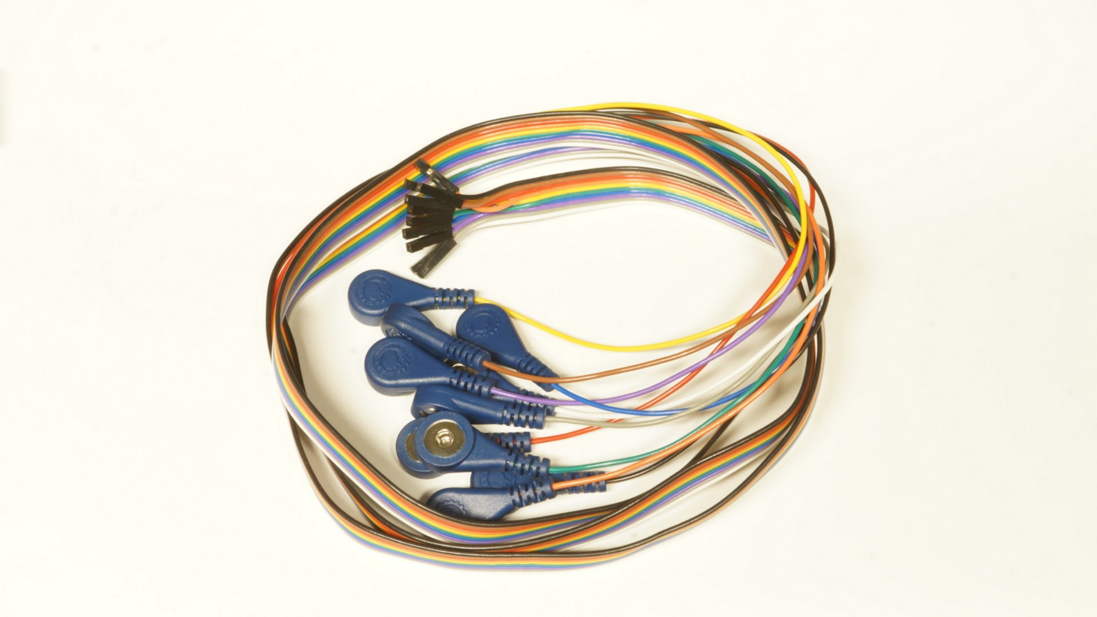


Figure 18. OpenBCI EMG/ECG Snap electrode cables

4. Signal Acquisition: The previously mentioned Python code was used to acquire the EEG signal from the electrodes with a sampling rate of 250 Hz. Such code utilizes BrainFlow and MNE, Python libraries that allows establish connection with the OpenBCI Cyton board and record EEG data which is saved as .edf and .fif formats for further preprocessing.
5. Task Paradigm.

The task consisted of 6 phases. The participant was required to keep its eyes closed during the entire task in order to avoid eye blink artifacts:

1. Participant was asked to relax. Since the EEG signal tends to fluctuate at the first 5 seconds of measurement, the recording started during relaxation phase until the signal became stable.
2. The participant was then requested to physically move the left hand. This had a double function: first it reminded the user the feelings of moving the hand for later motor imagery and secondly, physical movement was saved for possible further analysis.
3. After physical movement of left hand, the user was solicited to imagine moving his left hand.
4. The subject was then asked to relax. This phase can be used in future research for including relaxation in our study and classify 3 classes (left hand, right hand and relax) instead of the 2 classes being analyzed in this study (left hand and right hand).
5. In this phase, the user was asked to physically move its right hand.

6. Finally, the participant was requested to imagine moving its right hand.

The task had a variable length in both phases and trials since the length of each phase could be modified and it existed the possibility to repeat the phases in the same file.

There were timestamps obtained at the beginning and end of each phase so after each recording, different events were created to identify in which specific moment in the recording such phases happened.

The code employed for data acquisition allowed to get high-quality EEG information from the participant during motor imagery. This data was then used for subsequent data preprocessing and analysis.

In this research, several methods for connecting to the OpenBCI Cyton board for data acquisition were explored and assessed. One option involved using ROS through a publisher-subscriber mechanism for real-time streaming of EEG data that could be visualized and processed using Python code. Unfortunately, this approach proved unsuitable due to issues related to increasing latency as more data was streamed, making it impractical for the purposes at hand. In future work this issue might be addressed by utilizing RAPLET, a tool for evaluating ROS publisher-subscriber latency (Nishimura et al., 2021, pp. 41-42, 49-50).

Additionally, the LSL (Blum et al., 2021, pp. 3-4) in conjunction with Python library MNE-realtime was tested as an alternative option for real-time data interaction. Unfortunately, MNE-realtime offered limited functionality and options compared to BrainFlow and MNE libraries which ultimately were chosen for data acquisition.

3.4.2 Data Acquisition Parameters

These parameters were selected with care to ensure reliable and accurate data collection. During data acquisition, the following parameters were used:

1. Sampling rate: The EEG data were collected at 250 Hz.
2. Electrode placement: 3 electrodes were placed at C3, C4 and Cz positions.
3. Signal Averaging: 151 motor imagery events of each type were combined to achieve averaging. The averaged data were then used for further analysis.
4. Task Paradigm. The motor imagery tasks were based on the established protocols found in literature. This includes tasks like hand clenching and open-close movement imagination (K. Wang et al., 2017, pp. 668-669). Participant was told to follow the instructions and to maintain a relaxed state while data acquisition took place.
5. Recording Time: Each session lasted approximately 10-30 minutes depending on participant's fatigue and comfort. To minimize discomfort and ensure high quality data collection, sufficient rest breaks were provided.

These parameters were uniform across data acquisition sessions, ensuring standardization and reliability. As it can be seen in Figure 19, the output unfiltered EEG data coming from own measurements is full of noise and artifacts. Further filtering needs to be applied in order to clean the EEG data and improve signal quality.

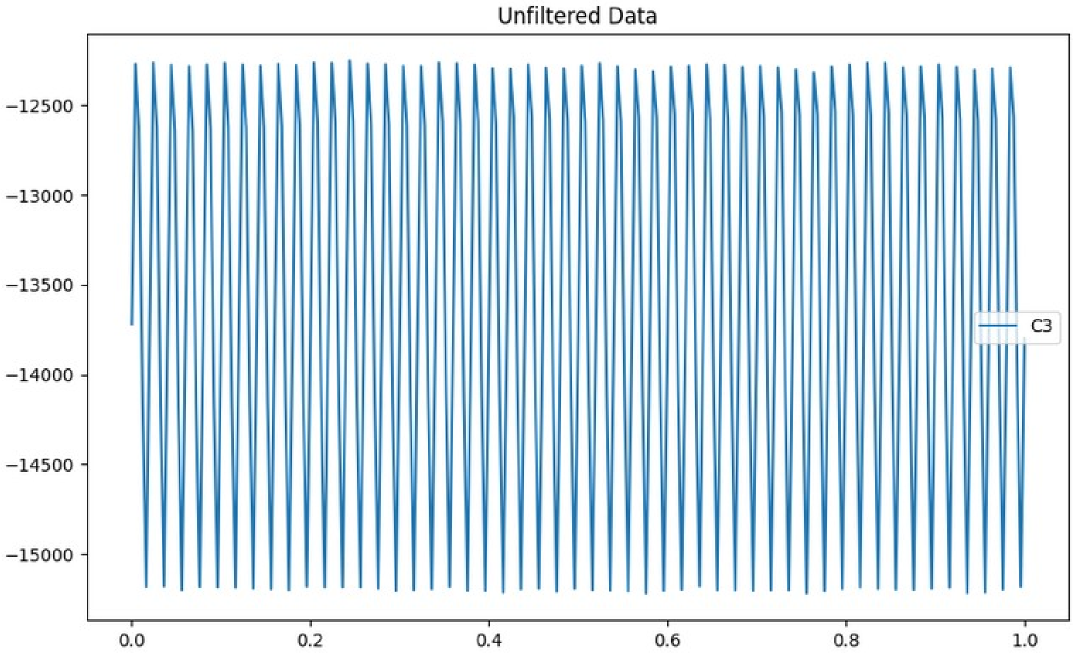


Figure 19. Unfiltered EEG data from data acquisition (only C3 channel)

3.4.3 Data Quality Assurance

It is essential to ensure that the data collected are accurate and reliable in order to obtain valid and reliable results. This study used the following measure to ensure the quality of data.

Before each measurement, data were visually inspected for any inconsistencies or artifacts using the OpenBCI GUI (version 5.1.0) software. OpenBCI GUI is a popular open-source software that allows you to acquire and analyze electrophysiological data from your brain, is well-known for its compatibility and flexibility with different types of BCI systems.

Finally, and with the objective to minimize the possibility of errors during data acquisition, proper electrode placement and calibration procedures were done during gathering sessions.

3.4.4 Data Management

To ensure that the data collected is secure, confidential, and accessible, data management was used. These steps were taken:

1. Data Storage: Raw data from both data acquisition and public datasets were securely stored on a password-protected computer that was only accessible to authorized personnel. To ensure data separation and avoid any data contamination, the data were stored in hierarchical folders.
2. Data Backup: To protect against data loss caused by hardware failures, or other unforeseeable circumstances, regular backups of the raw data were made and kept in a private GitHub repository.
3. Data Privacy: All data obtained during data acquisition as same as data from public dataset were assigned nonspecific codes or pseudonyms. In the case of own dataset, it was a single participant, and no privacy or confidentiality was compromised. Any personally identifiable information (PII) in either public dataset or own data was deleted or encrypted in accordance with applicable data privacy regulations.
4. Data documentation: This is a detailed description of data management procedures. It includes data storage, backups and privacy.

In general, data management procedures were strictly followed to ensure data integrity, security, and quality throughout the research process.

3.4.5 Data Acquisition Timeline

Below is a summary of the timeline for data acquisition:

1. Preparatory Phase: [16.12.2022 – 4.1.2023] The experimental protocol was designed for data collection and conducted some training/familiarization sessions mainly wearing the headset and adjusting for obtaining good quality signal. This phase involved the installation and configuration of all software and tools and the setting up of data acquisition equipment.
2. Pilot Data Collection: [5.1.2023 – 15.1.2023] A pilot data collection was done to validate the experimental protocol and verify that the data acquisition process was functioning properly. The data was carefully reviewed and analyzed to determine any problems or challenges before the main data collection.
3. Main Data Collection: [16.1.2023 – 20.2.2023] Following the successful completion of pilot data collection, main data collection was initiated. The study involved a single participant as previously stated. Data was collected using a Python code that combined BrainFlow and MNE libraries. To

minimize potential confounding factors, the data collection sessions were held in a controlled setting to ensure consistency across data.

4. Data Storage and Backup: [16.12.2022 - 20.2.2023] After data cleaning and preprocessing was completed, data were securely stored and backed up. Periodical backups of the data were made to protect data integrity and prevent data loss.

The data acquisition process was well managed and followed a defined timeline. All steps were taken to ensure that data integrity, security, and quality were maintained throughout the process.

3.4.5.1 Public Dataset: BCI Competition IV - Graz Data Set A

The BCI Competition IV - Graz Data Set A (Brunner et al., 2008) is a publicly accessible dataset containing EEG data from nine subjects recorded at Graz University of Technology, Austria as part of the BCI Competition IV. This experimental paradigm employed cue-based BCI with four motor imagery tasks: imagination of movement of left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4). Two sessions on different days were recorded for each subject with each consisting of 6 measurements separated by short breaks (48 trials per class), making for a total of 288 trials per session.

EEG data was recorded using 22 Ag/AgCl electrodes at an inter-electrode distance of 3.5 cm. The left mastoid served as the reference electrode and right mastoid served as ground electrode. Signals were sampled at 250 Hz and bandpass filtered between 0.5 Hz and 100 Hz, with an additional 50 Hz notch filter enabled to suppress line noise. Three monopolar EOG channels (Electrooculogram) channels were also recorded at 250 Hz with similar filters but must not be used for classification.

The dataset is stored in General Data Format (GDF), with one file per subject and session. However, only one session contains class labels for all trials while the other session serves to test and evaluate classifier performance. All files are listed below:

ID	Training File	Evaluation File
1	A01T.gdf	A01E.gdf
2	A02T.gdf	A02E.gdf
3	A03T.gdf	A03E.gdf
4	A04T.gdf	A04E.gdf
5	A05T.gdf	A05E.gdf
6	A06T.gdf	A06E.gdf

7	A07T.gdf	A07E.gdf
8	A08T.gdf	A08E.gdf
9	A09T.gdf	A09E.gdf

3.5 Data Augmentation

Raw EEG data was loaded from the input directory using the MNE-Python library. ICA was applied with “n_components” set to 8 and “random_state” set at 42, producing 8 independent components which could then be used to correct the original raw data using it to apply method in the ICA object. Finally, after applying corrections made by ICA correction, this augmented data was saved into a new file in a specific directory created for storing data generated through data augmentation.

ICA was chosen as the data augmentation technique in this study due to its capacity for decomposing multichannel EEG signals into their constituent independent components. As an unbiased source separation method, ICA assumes the observed signals are composed of linear mixtures of sources. By extracting these distinct brain activities or sources not directly visible in raw data, such as eye blinks or muscle artifacts (Motdhare & Mathur, 2022, pp. 87-90), ICA can identify distinct brain activities or sources not readily observable within raw measurements. By applying ICA for data augmentation, it effectively removed eye blinks and muscle artifacts from EEG recordings that often cause noise in these recordings. This improved the quality and diversity of the augmented data set by making it more representative of true brain activities.

3.6 Data Preprocessing

In this thesis, several preprocessing steps were applied to EEG data in order to guarantee its quality and suitability for training or classification applications. These included signal filtering, segmentation, time-frequency analysis, feature extraction, feature selection, data splitting, one-hot-encoding and electrode pair selection. These steps allowed to obtain meaningful information from EEG data.

3.6.1 EEG Signal Filtering

Filtering of raw EEG signals is a necessary step in preprocessing to guarantee data quality for motor imagery tasks (de Cheveigné & Nelken, 2019, p. 280). In this study, the raw EEG signals were filtered using both bandpass filtering and bandstop filtering with specific settings.

A bandpass filter was applied to preserve the frequency range of interest, set at 0.5-40 Hz. This range corresponded to brain activity studied in this thesis, including slow cortical potentials and beta/gamma oscillations associated with motor imagery tasks. By eliminating frequencies outside of this range that are not relevant for analysis, signal quality was improved using a Butterworth filter with an order of 4.

Additionally, a bandstop filter was employed to eliminate potential powerline interference at 50 Hz. This interference can occur due to electrical power supply problems in many countries and introduce noise into EEG signals (Zhang, S., et al., 2020, pp. 2-4, 7-11). Setting this filter to 49-51 Hz effectively attenuated any potential powerline noise present in EEG data. Furthermore, a Butterworth filter with an order of 3 was utilized as the bandstop filter. After the filtering step, EEG signal quality (Figure 20) is much cleaner and representative in comparison with unfiltered raw data (Figure 19).

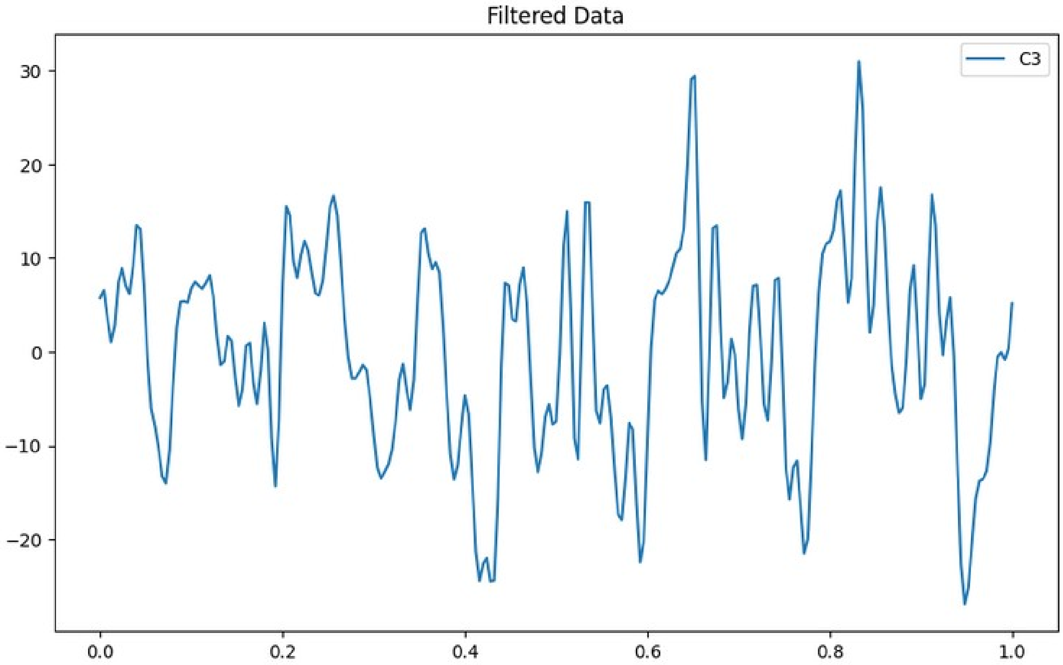


Figure 20. Filtered EEG data from data acquisition (only C3 channel)

3.6.2 Segmentation

In this study, EEG signals were preprocessed by first downsampling to a lower sampling rate and selecting only main channels for analysis. Common average reference (CAR) filtering was then utilized to minimize noise sources and increase signal-to-noise ratio (Tsuchimoto et al., 2021, p. 2).

Next, events of interest related to left and right hand motor imagery tasks were selected from annotations using a specific pattern. Epochs or time intervals of 6 seconds duration were extracted from filtered data around these events in order to ensure each epoch contained only one motor imagery task thereby reducing nonstationary and streamlining feature extraction and classification tasks for further analysis.

Reasons behind segmentation included reduced computational requirements and easier analysis. By breaking up the signal into specific periods, analysis could focus on specific features or patterns instead of an entire signal, improving accuracy and reliability while making extracting meaningful information from it more straightforward.

3.6.3 Time-Frequency Analysis

Time-Frequency Analysis was carried out using the tfr_multitaper function and multitaper spectral estimation method. The epoch data, which represents segmented EEG data, was passed to the function along with frequency range frequencies (from 2 to 45 Hz) and bins freq_bins (defined as alpha, low_beta, high_beta, and low_gamma ranges). The resulting time-frequency representation of this epoch data can then be further processed for feature extraction.

3.6.4 Feature Extraction

In this thesis, feature extraction involved calculating the spectral power of various frequency bands for each channel in EEG epoch time-frequency representation (Kim et al., 2018, pp. 2-6). Frequency bands were defined in advance and their respective spectral powers summed across all time points. Then, these values were stored as features within a DataFrame. This analysis produced a set of features representing the spectral power for each frequency band within an epoch data set for each channel in each data epoch, providing input for further analysis or classification tasks that help to understand the underlying patterns in the data (Singh & Krishnan 2023, p. 2).

3.6.5 Feature Selection

The features extracted from epoch data were stored into a DataFrame named feats_df. Here, each column represents the extracted features for each observation, such as spectral power values for different frequency bands and channels. Feats_df can be further refined using various feature selection techniques, depending on the analysis requirements. Options include filtering out low-variance features, using statistical tests or machine learning algorithms to select informative elements, and domain-specific feature selection methods.

3.6.6 Data Splitting

Once the data had been preprocessed, it was necessary to divide it into training, testing, and validation sets to evaluate its performance. To ensure the model could generalize well across a wider population, the dataset was split in an 80-10-10 ratio into training, test, and validation sets, using stratify parameter when necessary to maintain class distribution within split datasets when dealing with imbalanced classes.

3.6.7 One-Hot Encoding

In line with Zhang, Y. et al.'s (2020, p. 3) publication on machine learning, one-hot encoding was utilized to represent categorical variables. This technique created binary columns for each category, where 1 indicates presence and 0 implies absence. By doing so, it was easier to input categorical labels numerically into machine learning algorithms. Essentially, one-hot encoding is a way to convert categorical data into a numerical format that machine learning models can understand. Specifically, it was used the pd.get_dummies() function to transform the labels into one-hot encoding after splitting up the data into train, test, and validation sets.

3.6.8 Electrode Pair Selection

In this study, and since only 3 channels were used, all of them were selected. It is important to mention that, after electrode pair selection, the data was once again split into train, validation and test data, using the same ratio as previously mentioned (80:10:10). The training set was used to train the machine learning model, validation data was utilized for evaluating the performance of the model during its training, while the testing set evaluated its performance after training was completed.

Separating these three sets allowed to assess whether our model could generalize to unseen data when presented with new input data.

In conclusion, the preprocessing steps described in this section are essential for prepping EEG data for machine learning classification. All of which improve classification accuracy and reliability by properly prepping the EEG data for further analysis and interpretation.

3.7 Machine Learning

In this section, it is described the machine learning models utilized, including their structure, rationale for utilizing those, compilation and training process, detailing the hyperparameters used for classification of EEG signals. In this study it was employed a deep learning approach using CNN. Seven different binary classification CNN models were trained using features extracted from preprocessed EEG data. During training process, optimization took place by testing various hyperparameters. CNN models were trained with BCI Competition IV - Graz Data Set A. Each model was trained a at least five times to ensure that it was not negatively affected by randomly initialization (Amid et al., 2023, pp. 1-4). The DeepLearningProject repository on GitHub, created by James-Mc1ntyre, was used as inspiration for the implementation of some of the machine learning models.

Transfer learning was used to further improve model generalizability with new data. Pre-trained binary classification models were then fine-tuned using personal EEG data from own measurements in order to achieve superior results than models trained solely from public datasets. Different machine learning models have been tested for EEG signal classification in order to determine the best model for our study. Finally, this top-performing model was then used for motor imagery prediction for controlling a robot arm.

3.7.1 CNN Models

The CNN architecture consists of three main layers: the convolutional layer, pooling layer and fully connected layer. The convolutional layer extracts features from an input signal by performing convolution operations with learnable filters. The pooling layer reduces spatial dimension while maintaining important features. Finally, the fully connected layer performs classification by mapping extracted features to their corresponding class labels.

All models were trained for 250 epochs and batch size of 10, using categorical cross-entropy as the loss function, Adam optimizer, and a learning rate of 1e-4. The only exceptions were the HopefullNet model, which was trained for 100 epochs, and the FullyConnectedNet model, which was trained for 200 epochs with a batch size of 100 and used a learning rate of 1e-2. The accuracy metric was then used for models' evaluation during training.

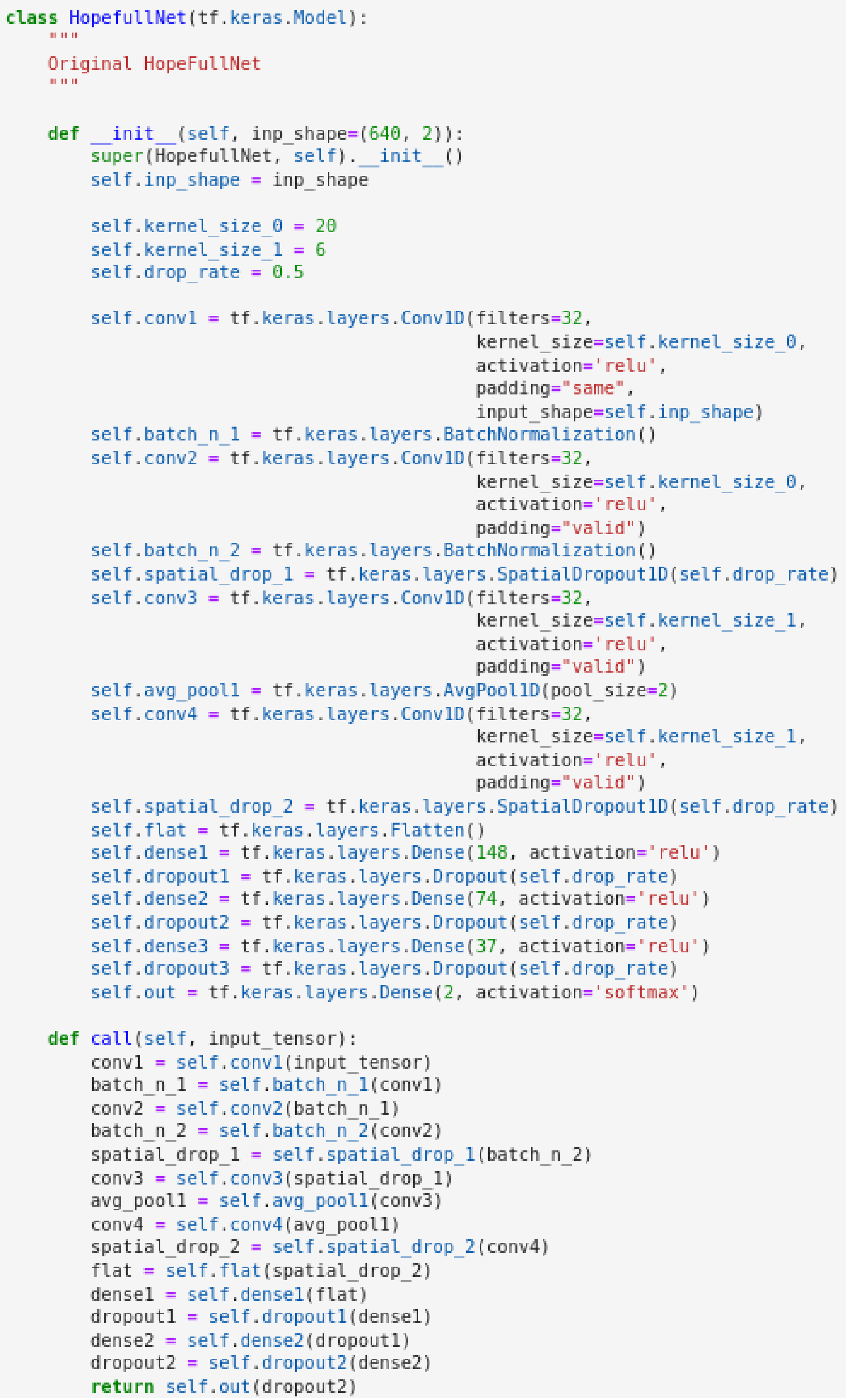
During training, the models utilized BCI Competition IV - Graz Data Set A (public dataset). The dataset was split into training data (used for training), validation (used during training for model optimization and performance evaluation) and test data (used for testing the performance of the trained model).

To ensure the models performance during training and avoid overfitting, both checkpoint and early stopping mechanisms were implemented. The checkpoint callback saved the model weights based on the validation loss, while the early stopping callback stopped training if there was no improvement in the validation loss in 4 consecutive epochs. To track progress more easily, verbosity was set to 1 for greater monitoring purposes.

PyTorch library was used to implement and train our CNN architectures using batches of preprocessed EEG signals with their associated class labels. After training, models' weights were saved for future references and models performance assessed through accuracy, precision, recall, and F1-score metrics. The models tested were the following:

3.7.1.1 HopefullNet Architecture, Rationale and Training

HopefullNet (Figure 21) is a deep learning model architecture composed of multiple convolutional layers, batch normalization layers, spatial dropout layers, average pooling layers, flatten layers and dense ones. The decision to use this model was based on its successful implementation in a related publication (Jia et al., 2022, pp. 8-12), which demonstrated high accuracy in multi-class classification. It was designed for classification tasks on input data with shape (640, 2) as input. The architecture includes four convolutional layers with different kernel sizes and activation functions followed by average pooling and spatial dropout layers to reduce overfitting. After these outputs are flattened, they pass through fully connected dense layers utilizing ReLU as activation function. Finally softmax activation is used for multiclass classification tasks.



```
class HopefullNet(tf.keras.Model):
    """
    Original HopeFullNet
    """

    def __init__(self, inp_shape=(640, 2)):
        super(HopefullNet, self).__init__()
        self.inp_shape = inp_shape

        self.kernel_size_0 = 20
        self.kernel_size_1 = 6
        self.drop_rate = 0.5

        self.conv1 = tf.keras.layers.Conv1D(filters=32,
                                            kernel_size=self.kernel_size_0,
                                            activation='relu',
                                            padding="same",
                                            input_shape=self.inp_shape)
        self.batch_n_1 = tf.keras.layers.BatchNormalization()
        self.conv2 = tf.keras.layers.Conv1D(filters=32,
                                            kernel_size=self.kernel_size_0,
                                            activation='relu',
                                            padding="valid")
        self.batch_n_2 = tf.keras.layers.BatchNormalization()
        self.spatial_drop_1 = tf.keras.layers.SpatialDropout1D(self.drop_rate)
        self.conv3 = tf.keras.layers.Conv1D(filters=32,
                                            kernel_size=self.kernel_size_1,
                                            activation='relu',
                                            padding="valid")
        self.avg_pool1 = tf.keras.layers.AvgPool1D(pool_size=2)
        self.conv4 = tf.keras.layers.Conv1D(filters=32,
                                            kernel_size=self.kernel_size_1,
                                            activation='relu',
                                            padding="valid")
        self.spatial_drop_2 = tf.keras.layers.SpatialDropout1D(self.drop_rate)
        self.flat = tf.keras.layers.Flatten()
        self.dense1 = tf.keras.layers.Dense(148, activation='relu')
        self.dropout1 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense2 = tf.keras.layers.Dense(74, activation='relu')
        self.dropout2 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense3 = tf.keras.layers.Dense(37, activation='relu')
        self.dropout3 = tf.keras.layers.Dropout(self.drop_rate)
        self.out = tf.keras.layers.Dense(2, activation='softmax')

    def call(self, input_tensor):
        conv1 = self.conv1(input_tensor)
        batch_n_1 = self.batch_n_1(conv1)
        conv2 = self.conv2(batch_n_1)
        batch_n_2 = self.batch_n_2(conv2)
        spatial_drop_1 = self.spatial_drop_1(batch_n_2)
        conv3 = self.conv3(spatial_drop_1)
        avg_pool1 = self.avg_pool1(conv3)
        conv4 = self.conv4(avg_pool1)
        spatial_drop_2 = self.spatial_drop_2(conv4)
        flat = self.flat(spatial_drop_2)
        dense1 = self.dense1(flat)
        dropout1 = self.dropout1(dense1)
        dense2 = self.dense2(dropout1)
        dropout2 = self.dropout2(dense2)
        return self.out(dropout2)
```

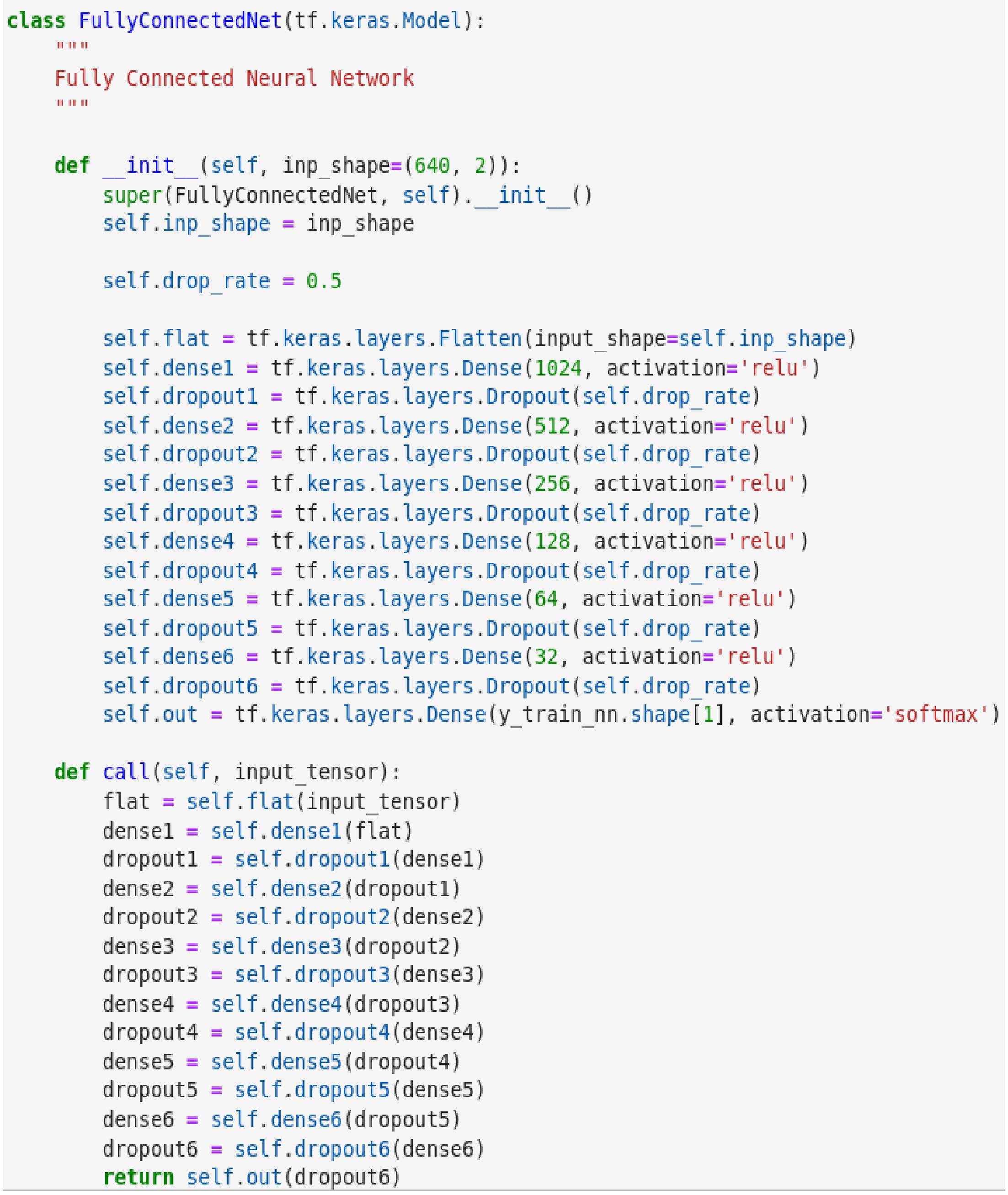
Figure 21. HopefullNet Model Architecture

HopefullNet was selected for this task due to its architecture's capacity to capture local and global patterns in input data through convolutional layers and pooling operations. Batch normalization and spatial dropout layers help improve generalization ability by reducing overfitting. Furthermore, fully connected dense layers enable the model to learn complex non-linear relationships within data, while softmax activation in the output layer makes it ideal for multi-class classification tasks.

3.7.1.2 FullyConnectedNet Architecture, Rationale and Training

FullyConnectedNet (Figure 22) is a neural network architecture built using TensorFlow's Keras library. It consists of multiple fully connected (dense) layers with ReLU activation functions followed by dropout layers to prevent overfitting. In total, six dense layers exist with decreasing numbers of neurons from 1024 to 32, as well as an output layer with softmax activation function for multiclass classification purposes. The input shape of the network is set to (640, 2) which represents its input data's dimensions.

FullyConnectedNet architecture consists of multiple dense layers, followed by a dropout layer. The dropout rate is set to 0.5, meaning that 50% of neurons in each dense layer are randomly dropped out during training to prevent overfitting. As more neurons join this network, its number decreases from 1024 to 32, enabling it to learn increasingly abstract representations of input data as it progresses through its layers. Finally, its final output has a softmax activation function which produces class probabilities for multiclass classification tasks.



```python
class FullyConnectedNet(tf.keras.Model):
    """
    Fully Connected Neural Network
    """

    def __init__(self, inp_shape=(640, 2)):
        super(FullyConnectedNet, self).__init__()
        self.inp_shape = inp_shape

        self.drop_rate = 0.5

        self.flat = tf.keras.layers.Flatten(input_shape=self.inp_shape)
        self.dense1 = tf.keras.layers.Dense(1024, activation='relu')
        self.dropout1 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense2 = tf.keras.layers.Dense(512, activation='relu')
        self.dropout2 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense3 = tf.keras.layers.Dense(256, activation='relu')
        self.dropout3 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense4 = tf.keras.layers.Dense(128, activation='relu')
        self.dropout4 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense5 = tf.keras.layers.Dense(64, activation='relu')
        self.dropout5 = tf.keras.layers.Dropout(self.drop_rate)
        self.dense6 = tf.keras.layers.Dense(32, activation='relu')
        self.dropout6 = tf.keras.layers.Dropout(self.drop_rate)
        self.out = tf.keras.layers.Dense(y_train_nn.shape[1], activation='softmax')

    def call(self, input_tensor):
        flat = self.flat(input_tensor)
        dense1 = self.dense1(flat)
        dropout1 = self.dropout1(dense1)
        dense2 = self.dense2(dropout1)
        dropout2 = self.dropout2(dense2)
        dense3 = self.dense3(dropout2)
        dropout3 = self.dropout3(dense3)
        dense4 = self.dense4(dropout3)
        dropout4 = self.dropout4(dense4)
        dense5 = self.dense5(dropout4)
        dropout5 = self.dropout5(dense5)
        dense6 = self.dense6(dropout5)
        dropout6 = self.dropout6(dense6)
        return self.out(dropout6)
```

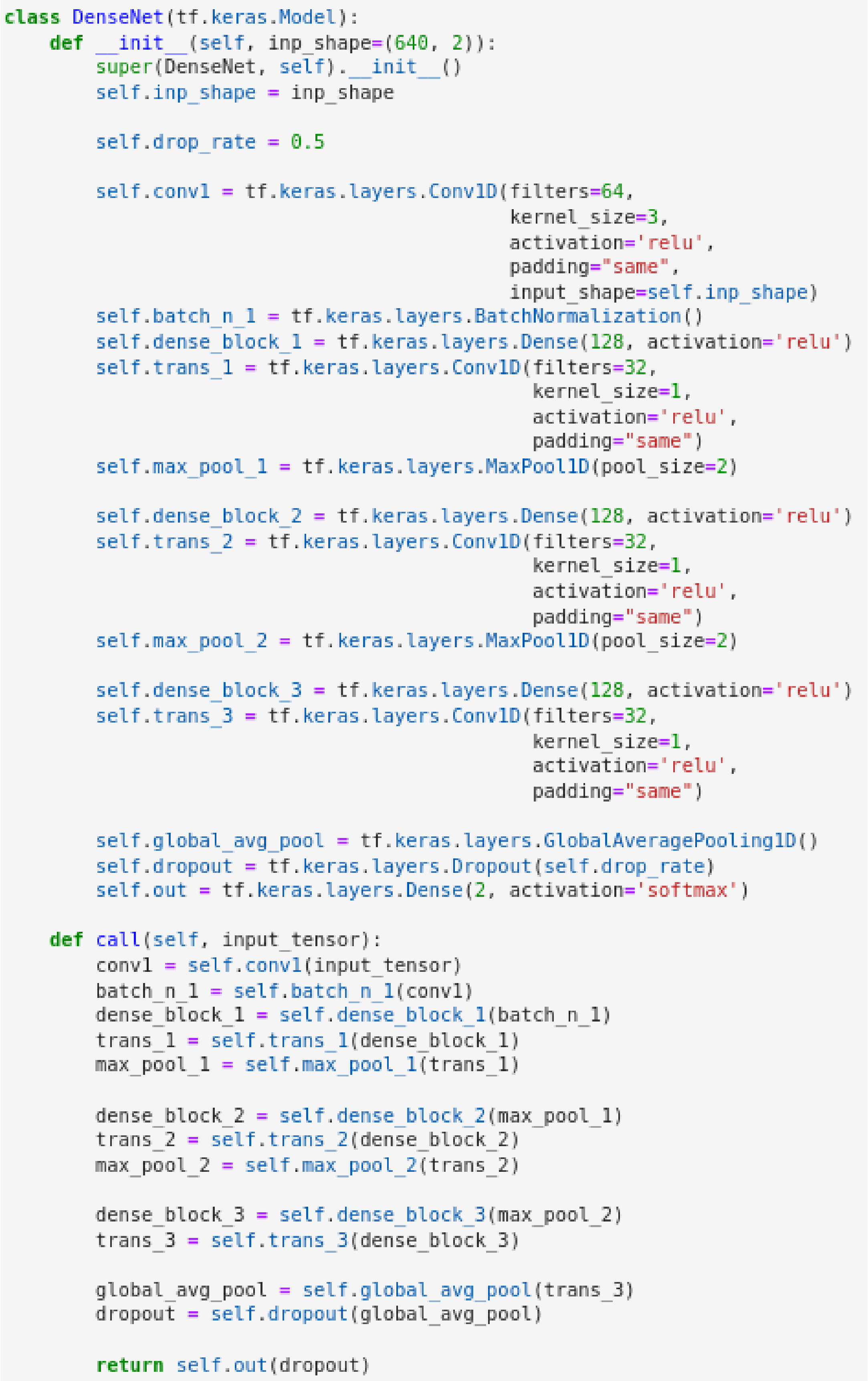
Figure 22. FullyConnectedNet Model Architecture

For this task, the FullyConnectedNet architecture was selected due to its capacity for learning complex patterns from input data using multiple dense layers. The ReLU activation function helps mitigate the vanishing gradient problem while dropout layers provide regularization to prevent overfitting. Moreover, with fewer neurons per dense layer, the network can learn hierarchical representations of input data - beneficial for image classification tasks.

3.7.1.3 DenseNet Architecture, Rationale and Training

DenseNet is a CNN architecture designed by Huang et al. in 2016 for "Densely Connected Convolution Networks." This unique approach uses layers that receive input not only from their preceding layer, but also all previous ones; allowing for efficient feature reuse and gradient flow with improved accuracy and parameter efficiency when compared to traditional CNNs.

DenseNet architecture utilized in this study (Figure 23) is a simplified version that does not contain skip connections between dense blocks. This model utilizes three dense blocks, each composed of multiple convolutional layers with a growth rate of 128 and followed by a 1x1 convolutional for compression. Each dense block is connected to a transition layer which includes one 1x1 convolutional for dimension reduction and max pooling for downsampling. A global average pooling layer then serves to further reduce spatial dimension before the final fully connected softmax layer for classification. A dropout rate of 0.5 is applied to this global average pooling layer to prevent overfitting.



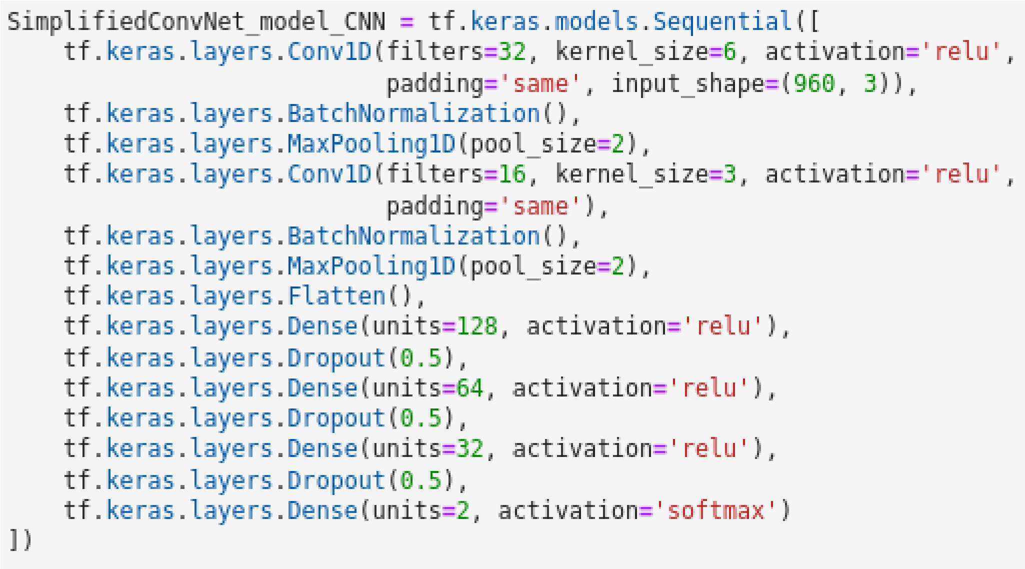
```
class DenseNet(tf.keras.Model):
    def __init__(self, inp_shape=(640, 2)):
        super(DenseNet, self).__init__()
        self.inp_shape = inp_shape

        self.drop_rate = 0.5

        self.conv1 = tf.keras.layers.Conv1D(filters=64,
                                            kernel_size=3,
                                            activation='relu',
                                            padding="same",
                                            input_shape=self.inp_shape)
        self.batch_n_1 = tf.keras.layers.BatchNormalization()
        self.dense_block_1 = tf.keras.layers.Dense(128, activation='relu')
        self.trans_1 = tf.keras.layers.Conv1D(filters=32,
                                              kernel_size=1,
                                              activation='relu',
                                              padding="same")
        self.max_pool_1 = tf.keras.layers.MaxPool1D(pool_size=2)

        self.dense_block_2 = tf.keras.layers.Dense(128, activation='relu')
        self.trans_2 = tf.keras.layers.Conv1D(filters=32,
                                              kernel_size=1,
                                              activation='relu',
                                              padding="same")
        self.max_pool_2 = tf.keras.layers.MaxPool1D(pool_size=2)

        self.dense_block_3 = tf.keras.layers.Dense(128, activation='relu')
        self.trans_3 = tf.keras.layers.Conv1D(filters=32,
                                              kernel_size=1,
                                              activation='relu',
                                              padding="same")

        self.global_avg_pool = tf.keras.layers.GlobalAveragePooling1D()
        self.dropout = tf.keras.layers.Dropout(self.drop_rate)
        self.out = tf.keras.layers.Dense(2, activation='softmax')

    def call(self, input_tensor):
        conv1 = self.conv1(input_tensor)
        batch_n_1 = self.batch_n_1(conv1)
        dense_block_1 = self.dense_block_1(batch_n_1)
        trans_1 = self.trans_1(dense_block_1)
        max_pool_1 = self.max_pool_1(trans_1)

        dense_block_2 = self.dense_block_2(max_pool_1)
        trans_2 = self.trans_2(dense_block_2)
        max_pool_2 = self.max_pool_2(trans_2)

        dense_block_3 = self.dense_block_3(max_pool_2)
        trans_3 = self.trans_3(dense_block_3)

        global_avg_pool = self.global_avg_pool(trans_3)
        dropout = self.dropout(global_avg_pool)

        return self.out(dropout)
```

Figure 23. DenseNet Model Architecture

DenseNet was chosen for its ability to capture long-range dependencies in input data due to its dense connectivity, leading to improved accuracy and parameter efficiency compared to traditional CNN architectures. A growth rate of 128 was chosen after experimentation to strike a balance between model complexity and performance. Utilizing transition layers with 1x1 convolutional layers and max pooling helps reduce spatial dimensions as well as control model complexity.

3.7.1.4 SimplifiedConvNet Architecture, Rationale and Training

The architecture of the SimplifiedConvNet model (Figure 24) consists of multiple layers, such as Convolutional layers, Batch Normalization layers, Flatten layer, Dense layers and Dropout ones. It begins with a Conv1D layer with 32 filters and a kernel size of 6 with an activation function called 'relu'. Following that is a Batch Normalization layer and MaxPooling1D layer with a pool size of 2. Afterwards, another Conv1D layer with 16 filters, kernel size 3, and 'relu' activation function follows, followed by another Batch Normalization and MaxPooling1D with its pool size set at 2. The Flatten layer is used to transform data into a 1D vector and pass through Dense layers with different units (128, 64, and 32) using the 'relu' activation function. Dropout layers with an effective dropout rate of 0.5 are added after each Dense layer to avoid overfitting. Finally, an output layer with units equal to the number of classes in the target data and using 'softmax' activation function produces predicted class probabilities.



```
SimplifiedConvNet_model_CNN = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(filters=32, kernel_size=6, activation='relu',
                           padding='same', input_shape=(960, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Conv1D(filters=16, kernel_size=3, activation='relu',
                           padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(units=128, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=64, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=32, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=2, activation='softmax')
])
```

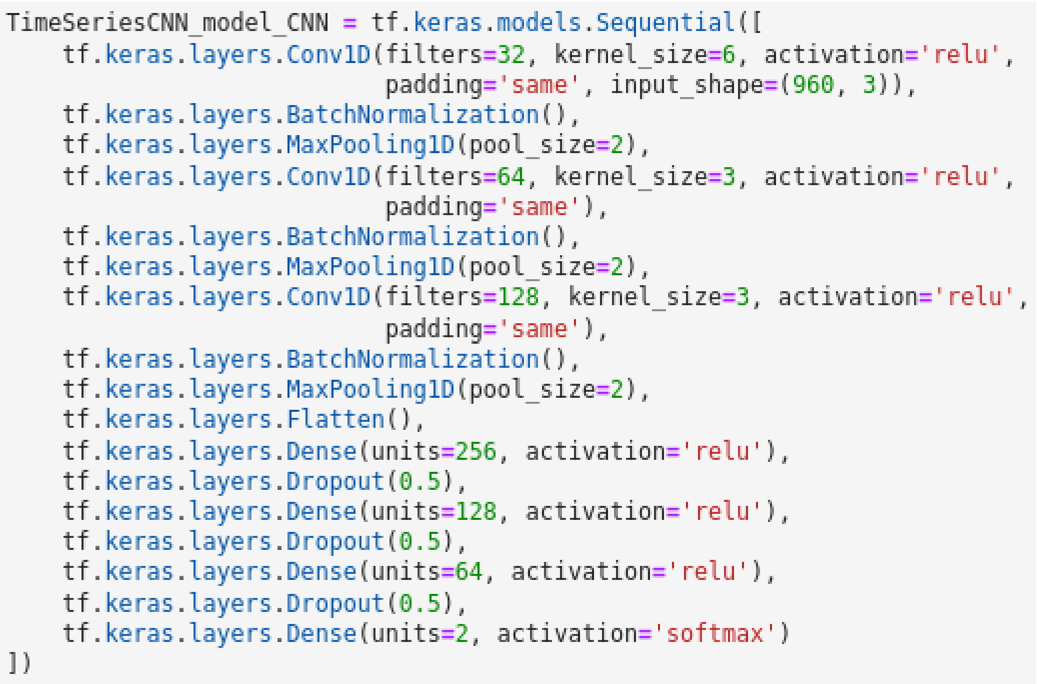
Figure 24. SimplifiedConvNet Model Architecture

The SimplifiedConvNet model architecture was chosen due to the need for a simpler model with fewer parameters and greater computational efficiency. The model includes multiple Conv1D layers with different filter sizes to capture different patterns in data, followed by Batch Normalization layers which normalize inputs and speed up training time. Dropout layers reduce overfitting while Dense units with smaller units reduce model complexity. This architecture should provide an optimal balance between model complexity and performance.

3.7.1.5 TimeSeriesCNN Architecture, Rationale and Training

The TimeSeriesCNN model architecture (Figure 25) is a CNN implemented using the Keras library of TensorFlow. It consists of several layers, such as Conv1D with various filter sizes, BatchNormalization for normalization, MaxPooling1D for downsampling, Flatten for flattening the output, and Dense with various units and activation functions. Dropout layers are added to prevent overfitting during training.

The architecture begins with a Conv1D layer with 32 filters and a kernel size of 6, followed by BatchNormalization and MaxPooling1D layers. Thereafter, two additional Conv1D layers follow with 64 and 128 filters respectively, both followed by BatchNormalization and MaxPooling1D layers. After flattening the output, three Dense layers with different units (256, 128, 64) as well as ReLU activation functions are added before dropping out at 0.5 percent. Finally, an output layer has units equal to the number of classes in the target variable plus softmax activation function for multiclass classification purposes.



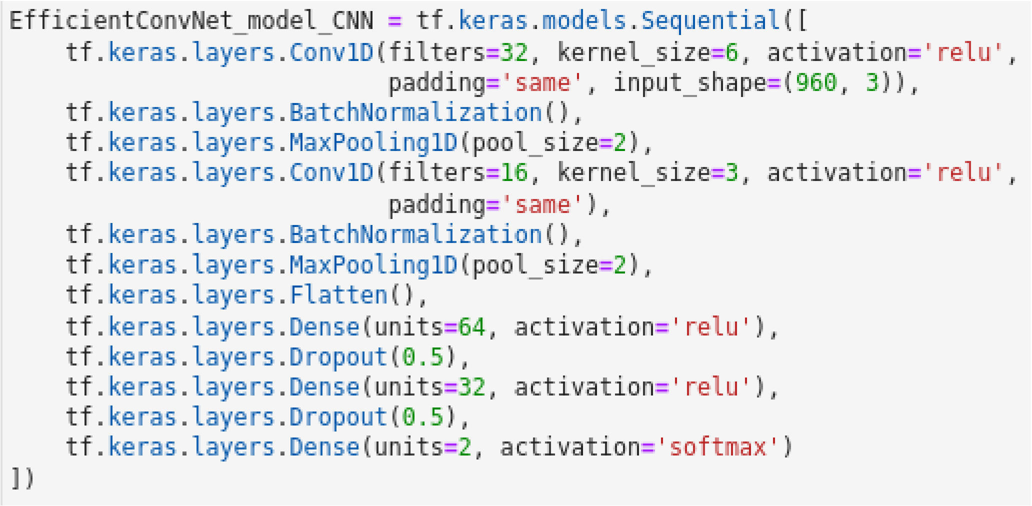
```
TimeSeriesCNN_model_CNN = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(filters=32, kernel_size=6, activation='relu',
                           padding='same', input_shape=(960, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Conv1D(filters=64, kernel_size=3, activation='relu',
                           padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Conv1D(filters=128, kernel_size=3, activation='relu',
                           padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(units=256, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=128, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=64, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=2, activation='softmax')
])
```

Figure 25. TimeSeriesCNN Model Architecture

The TimeSeriesCNN model architecture was chosen based on its ability to capture relevant features from input data, particularly time-series data like this study's. Conv1D layers have long been known for processing sequences and patterns in time series data; adding BatchNormalization and Dropout layers helps with regularization while avoiding overfitting. The architecture is deep enough to capture complex patterns but not too complex that it becomes overfitting.

3.7.1.6 EfficientConvNet Architecture, Rationale and Training

The EfficientConvNet model architecture (Figure 26) is defined using TensorFlow-Keras' Sequential API. It consists of several layers, such as Conv1D, BatchNormalization, MaxPooling1D, Flatten, Dense and Dropout. Conv1D layers are employed for one-dimensional convolutional operations, featuring various filter sizes and activation functions. BatchNormalization normalizes the inputs to each layer, while MaxPooling1D downsamples data. The Flatten layer is used to transform the output from convolutional layers into a 1D vector, which then passes through fully connected Dense layers. Dropout helps with regularization to prevent overfitting during training. Finally, this final Dense layer utilizes a softmax activation function for multi-class classification with probabilities.



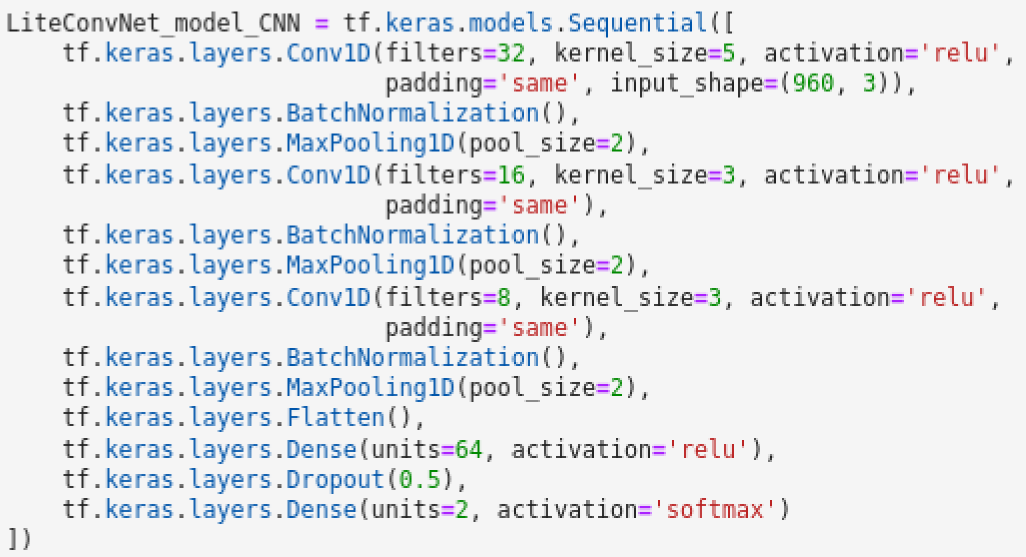
```
EfficientConvNet_model_CNN = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(filters=32, kernel_size=6, activation='relu',
                           padding='same', input_shape=(960, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Conv1D(filters=16, kernel_size=3, activation='relu',
                           padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(units=64, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=32, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=2, activation='softmax')
])
```

Figure 26. EfficientConvNet Model Architecture

The EfficientConvNet model was selected due to its balance between model complexity and performance. It has a moderate number of layers and parameters which reduces the risk of overfitting, making it suitable for datasets with limited training samples. Furthermore, its architecture was engineered to capture relevant features from input data while still remaining computationally efficient.

3.7.1.7 LiteConvNet Architecture, Rationale and Training

The LiteConvNet model (Figure 27) is a CNN architecture composed of multiple layers. It starts with a 1D convolutional layer with 32 filters, 5 kernel sizes, ReLU activation followed by batch normalization and max pooling with 2 pool sizes. After this comes another 1D convolutional layer with 16 filters and a kernel size of 3, as well as ReLU activation. Finally, batch normalization and max pooling are applied before concluding this step. Finally, there is a 3D convolutional layer with 8 filters and 3 kernel size that utilize ReLU activation followed by batch normalization and max pooling. The output from these convolutional layers is flattened and connected to a fully connected dense layer with 64 units and ReLU activation; additionally, an extra dropout layer at rate 0.5 prevents overfitting. Finally, an output layer has units equal to the number of classes in the target variable with softmax activation for multi-class classification applications.



```
LiteConvNet_model_CNN = tf.keras.models.Sequential([
    tf.keras.layers.Conv1D(filters=32, kernel_size=5, activation='relu',
                           padding='same', input_shape=(960, 3)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Conv1D(filters=16, kernel_size=3, activation='relu',
                           padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Conv1D(filters=8, kernel_size=3, activation='relu',
                           padding='same'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling1D(pool_size=2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(units=64, activation='relu'),
    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Dense(units=2, activation='softmax')
])
```

Figure 27. LiteConvNet Model Architecture

The LiteConvNet model architecture was chosen through experimentation and tuning of hyperparameters. It is designed to be simpler than previous models, with fewer convolutional layers and lower filter numbers, in order to reduce computational complexity and potential overfitting. Furthermore, batch normalization and dropout layers help improve model generalization while preventing overfitting.

3.7.2 Transfer Learning

Transfer learning is a powerful technique that can enhance the performance of machine learning models when there is limited training data. In this study, transfer learning was applied to improve CNN model performance at generalizing with new data. All previously saved pre-trained CNN models served as the starting point and were retrained using the same CNN models with EEG data obtained during data acquisition. Finally, the trained models were evaluated on an actual set of EEG data.

Each model was trained for 100 epochs of training data with a batch size of 32. During training, validation data was used for model evaluation, weights were saved at the best performing epoch based on validation loss, and an early stopping strategy of 4 epochs was employed to prevent overfitting.

3.8 Robot Control using ROS

Robot Operating System (ROS) is an open-source software framework widely utilized in robotics. ROS was utilized in this project to control an Elfin5 robot arm through simulation. The ROS MoveIt package was utilized to enable efficient and precise control of the Elfin5 robot arm by integrating the best pretrained model from the seven CNN models that were previously trained with a public dataset and retrained with personal data from data acquisition using transfer learning. This model, using signals received through an EEG headset, classified motor imagery from left and right hand before executing predefined trajectory for the robot arm depending on the prediction results.

3.8.1 ROS MoveIt Package

The ROS MoveIt package, a powerful motion planning framework, was utilized in this project to enable precise and efficient robot arm control. MoveIt provided the necessary tools necessary for planning and executing robot arm movements in a precise and efficient manner (Hernandez-Mendez et al., 2017, p. 1). It facilitated collision, detection, trajectory generation and path optimization tools allowed for easy design and testing of complex robotic tasks. The MoveIt visualization tools made complex robotic tasks much simpler to design and test. In this project, the ROS MoveIt package was utilized to control the robot arm movements based on the motor imagery classification obtained from the EEG data.

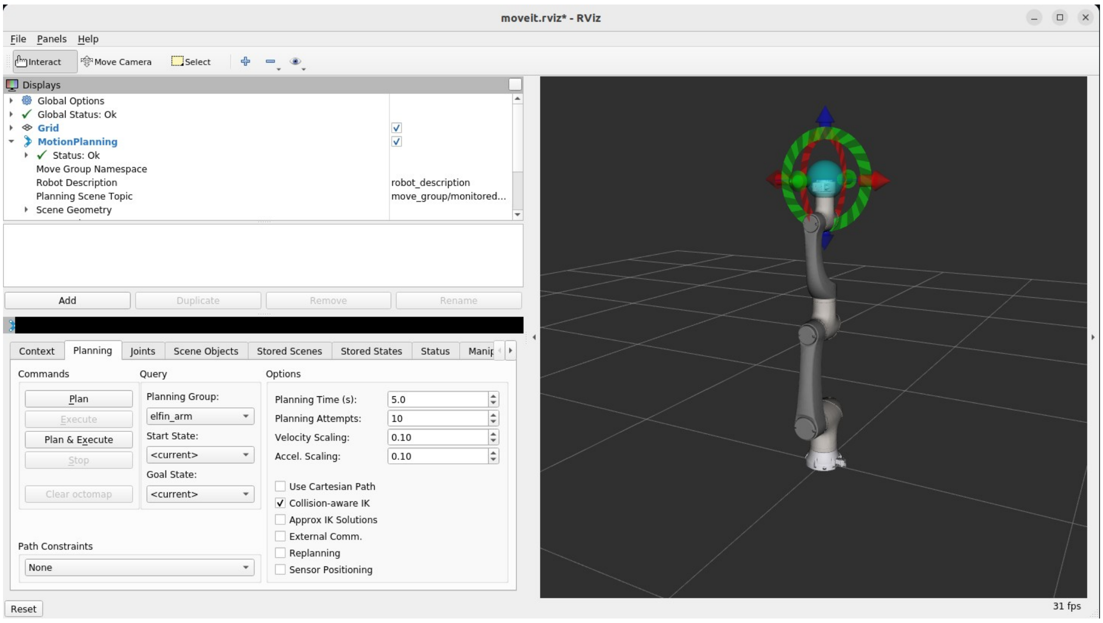
3.8.2 Elfin5 Robot Arm

The Elfin5 Robot Arm (Figure 28), a collaborative robot arm manufactured by Shenzhen Han's Robot Co., Ltd. was chosen for simulation purposes in this project. With its six degrees of freedom and an up to 5 kg payload capacity, this robot arm is an excellent device for performing assembly, material handling and quality inspection tasks. It was integrated into the ROS environment for controlled simulation of its movements using ROS Visualization (Rviz).

Figure 28. Elfin5 robot arm (Hans Robot, 2023)

3.8.3 Movement Task Design

The movement task was developed as a pick-and-place exercise, wherein user was instructed to imagine either left hand motor imagery for the robot arm to pick an object from the left side or imagine right hand motor imagery for the robot arm to pick the object from the right side. The movement task of the robot was executed in a simulated environment utilizing Rviz ROS tool as it can be seen in Figure 29.



Global Status: Ok

Status: Ok

Move Group Namespace

Robot Description: robot_description

Planning Scene Topic: move_group/monitored...

Planning Group: elfin_arm

Start State: <current>

Goal State: <current>

Path Constraints: None

Planning Time (s): 5.0

Planning Attempts: 10

Velocity Scaling: 0.10

Accel. Scaling: 0.10

☐ Use Cartesian Path

☑ Collision-aware IK

☐ Approx IK Solutions

☐ External Comm.

☐ Replanning

☐ Sensor Positioning

Figure 29. Elfin5 robot arm visualized in simulation using Rviz ROS tool

This task consisted of three stages.

1. Resting State: To provide an initial baseline for EEG recording, user was instructed to relax and remain still without moving or visualizing motor imagery for a set period of time. During this stage the user remained still and completely relax without making any movements or having motor imaginary visualization before starting EEG recordings.
2. EEG Measurement Stage: At this point, user was instructed to imagine left or right hand motor imagery while EEG data was acquired and preprocessed. Once preprocessed, this EEG data was run through a trained CNN model called TimeSeriesCNN, which provided optimal results when it came to predicting both public dataset and personal EEG data.
3. Classification and Robot Arm Movement Stage: At this stage, motor imagery was classified using the trained TimeSeriesCNN model. After that, Elfin5 Robot Arm in simulation would execute a joint space trajectory with its end effector moving to either left or right depending on its classification. An object (a cube) would then be created in the simulation, attached to the robot arm's end effector, and moved towards its center using cartesian space trajectory. Finally, this object would be detached and the robot arm would return back to its initial position, waiting for the next prediction.

3.9 Justification of Methods

The methods chosen in this thesis are justified by their effectiveness and compatibility with the goal of controlling the robot arm using EEG signals. EEG signals enable direct and noninvasive measurement of brain activity, making it suitable for capturing motor imagery patterns. Deep learning techniques, especially convolutional neural networks (CNNs), are used to automatically extract relevant features and accurately classify these patterns (Mao et al., 2020, p. 5).

The training process consists of seven CNN models initially trained with a public dataset and then retrained with own measurements using transfer learning. This approach leverages the models' general knowledge and individual-specific characteristics, increasing their ability to adapt to user’s motor imagery patterns (Zhang et al., 2021, p. 7). Transfer learning uses pre-trained weights from public datasets to capture features associated with motor imagery. Fine-tuning with own dataset allows the models to capture the unique nuances of each individual’s brain signals (Tibrewal et al., 2022, p. 13).

Integration with the ROS MoveIt package provides a robust framework for controlling the robot arm. MoveIt's motion planning capabilities, collision detection, trajectory generation, and path optimization tools make the Elfin5 robotic arm move more accurately and smoothly. By executing predefined trajectories based on motor imagery classification results, the system achieves accurate control of the robot arm.

4 RESULTS

4.1 3D Scanning, EEG Headset and Electrodes performance evaluation

Performance evaluation of EEG headset and electrodes was performed to assess their effectiveness in terms of comfort, fit, ease of use and signal quality. The evaluation results provided valuable insights into the performance of these components, which are crucial for obtaining reliable EEG data.

The electrodes utilized in this study were evaluated for their resistance, impedance and signal quality. Results demonstrated variations in the performance among different designs, materials and coatings used on electrodes.

4.1.1 3D Scanning Results

After testing multiple 3D scanners, Revopoint Mini scanner with Revopoint Studio software faced challenges in keeping track of the head due to dark color hair which is troublesome due to difficulties in capturing details and textures. As a result, the Revopoint Mini scanner was not suitable for the project. Kinect 3D with Skanect software was discarded due to its poor accuracy. Finally, Creality CR-Scan Lizard 3D scanner integrated with CR-Studio software was used based on the high-quality resulting model.

To overcome the challenges presented by dark-colored hair, a white swimming cap was utilized during scanning, enabling for an optimal scan. Unfortunately, some small wrinkles appeared in the final model due to the swimming cap not perfectly conforming to the head shape. These wrinkles needed to be addressed later through further smoothing steps as explained in methodology.

4.1.2 Headset Evaluation and Results

The evaluation of the 3D printed headset produced highly positive results in terms of comfortability, fit, and ease of use. The headset was found to be very lightweight, making long periods of use comfortable without feeling intrusive. Its flexibility allowed for a snug fit, contouring to the shape of the head, which further enhanced its comfort. Careful consideration of the head shape during printing process ensured an optimum fit for the user.

Importantly, the signal quality of the EEG recordings obtained using this headset was not negatively affected. Despite the lightweight and flexible design, the headset maintained outstanding signal quality indicating its suitability for accurate and reliable data collection. This underlines its utility as a user-friendly EEG recording solution suitable for successful use in research or clinical environments.

4.1.3 Electrodes Evaluation and Results

Evaluating the electrodes used in EEG recordings is a critical aspect of obtaining reliable and accurate data. Electrode performance, especially impedance and signal quality, has an enormous effect on EEG recordings and subsequent data analysis. In this section, the evaluation and results of the electrodes used in the study are presented. The performance of different electrode designs, materials, and coatings are assessed, providing insights into their suitability for EEG recordings.

It's worth noting that in the following tables, Electric Paint A is Electrically Conductive Paint, while Electric Paint B is Bare Conductive Electric Paint.

The findings of this evaluation contribute to the understanding of electrode performance in the context of EEG recordings and can guide future research and clinical applications.

The results were compared with a commercial EEG headset and electrode set. Results revealed that while the designed EEG headset and electrode system could produce signals with similar quality to the commercial system, it proved significantly more comfortable and user-friendly. This is a significant advance over existing commercial options, promising to enhance both reliability and comfort when using EEG-based robot control.

4.1.3.1 Electrodes Resistance Results

The resistance of different electrode designs, materials, and coatings were evaluated in this study. Results are summarized in Table 1, which shows that Headpin flat electrode design had the lowest overall resistance values among the three designs tested. When considering materials and coatings, Headpin Spiderman printed with Conductive Filaflex without any coating showed the highest resistance levels, while Siraya Tech Blu resin printed with Bare Conductive Electric Paint coating showed the lowest resistance values.

Table 1. Resistance Results by Electrode Design and Material/Coating (Resistance values are reported in Ohms and represent the average of multiple measurements)

Electrodes design / Material and Coating	Siraya Tech Blu resin (electric paint B)	Resione F80 resin (with electric paint B)	Conductive Filaflex	Conductive Filaflex (with electric paint A)	Conductive Filaflex (with electric paint B)	Original Enobio
Headpin flat	42.33	450	267.25	133.0	139.0	0.8
Headpin multiplestuds	70.33	927.5	610.0	266.0	41.33	0.88
Headpin Spiderman	60.67	460.0	1011.2	276.0	62.67	

4.1.3.2 Electrodes Impedance Results

In this section, impedance results are presented for various electrode designs and materials as well as different coating options. These results offer valuable insights into the impedance performance of various electrode configurations and can aid in selecting suitable designs and materials for EEG applications. Note that lower impedance values generally indicate better electrode performance, as they result in improved signal quality and reduced noise interference. The Table 2 summarizes impedance results for the tested electrode designs, materials and coatings and shows that Headpin flat had the lowest overall impedance values among all three electrode designs. In terms of materials and coatings, Headpin flat electrode printed with Conductive Filaflex without any coating showed the lowest impedance values versus multiplestuds printed with Resione F80 resin coated with Bare Conductive Electric Paint which showed the highest impedance value. Considering the overall performance of all three electrode designs, Conductive Filaflex without coating showed the best impedance performance, followed by Resione F80 resin with Bare Conductive Electric Paint coating.

Table 2. Impedance Results by Electrode Design and Material/Coating (Impedance values are reported in Ohms and represent the average of multiple measurements)

Electrodes design / Material and Coating	Siraya Tech Blu resin (electric paint B)	Resione F80 resin (with electric paint B)	Conductive Filaflex	Conductive Filaflex (with electric paint A)	Conductive Filaflex (with electric paint B)	Original Enobio
Headpin flat	647.5	22.14	16.5	12.0	30.5	439.66
Headpin multiplestuds	2984.0	4463.5	4393.33	3780.0	4139.33	59.33
Headpin Spiderman	3418.67	1808.0	1950.67	982.33	2188.0	

4.1.3.3 Signal Quality Results

Signal quality is a critical aspect of electrodes' performance (Radüntz, 2018, pp. 1-2). As earlier explained in methodology, signal quality results were ranked from 1 (indicating very good signal quality) to 5 (indicating very bad signal quality). These results were displayed for various electrode designs, materials and coatings, taking into consideration noise and artifacts present during signal visualization. The results of the signal quality of each electrode design, material and coating are summarized in Table 3, where Siraya Tech Blu resin with Bare Conductive Electric Paint and Conductive Filaflex with Electrically Conductive Paint have generally shown lower values in the ranking (indicating better signal quality) when compared with other materials and coatings. However, Conductive Filaflex without any coating scored higher (indicating worse signal quality) compared to the other configurations.

Table 3. Signal Quality Results by Electrode Design and Material/Coating

Electrodes design / Material and Coating	Siraya Tech Blu resin (electric paint B)	Resione F80 resin (with electric paint B)	Conductive Filaflex	Conductive Filaflex (with electric paint A)	Conductive Filaflex (with electric paint B)	Original Enobio
Headpin flat	2	4	5	2	3	1
Headpin multiplestuds	2	4	5	2	3	1
Headpin Spiderman	2	4	5	2	3	

4.1.3.4 Selection of Best Electrodes

Based on the results obtained from the resistance, impedance, and signal quality measurements of different electrode designs and material/coating combinations, the Headpin flat electrode printed with Conductive Filaflex without any coating appears to be the best choice for areas without hair, considering its low resistance and impedance values, indicating good electrical conductivity and reduced signal interference, even though the signal quality was relatively poor.

However, it's essential to keep in mind that the best electrode choice depends on both application and placement location. In this study, EEG channel positions are C3, Cz and C4, which are areas with hair. That the Headpin flat electrode may not be an appropriate choice due to hair affecting in the contact between electrode and skin and signal quality. Alternative designs and material/coating combinations need to be considered instead.

Based on previous information regarding to resistance, impedance and signal quality as well as considering flexibility as an element for improved comfort, the Headpin Spiderman electrode with Conductive Filaflex with Electrically Conductive Paint coating (Figure 30) seems like the optimal option for this research. This decision can be explained as follows:

1. Resistance: The Headpin Spiderman electrode printed with Conductive Filaflex with Electrically Conductive Paint as coating demonstrated a resistance value of 276.0, which is significantly lower compared to other combinations and the average resistance (378.57), indicating good electrical conductivity.

2. Impedance: When considering only electrodes without flat designs, the Headpin Spiderman electrode printed with Conductive Filaflex with Electrically Conductive Paint as coating showed the lowest impedance value (982.33). This indicates reduced impedance to electrical signal transmission between electrodes and skin surfaces, leading to better signal quality overall.

3. Signal Quality: The Headpin Spiderman electrode printed with Conductive Filaflex and Electrically Conductive Paint as coating achieved the best signal quality results. Siraya Tech Blu resin with Bare Conductive Electric Paint was equally successful, which confirms its excellent resistance and impedance results. Regardless, both performed reasonably well when considering their presence of hair in targeted EEG channel positions (C3, Cz and C4).

Additionally, the Headpin Spiderman electrode design with multiple spikes and flexibility may provide a better contact between electrode and skin surface, helping overcome potential interference from hair while increasing overall electrode performance.

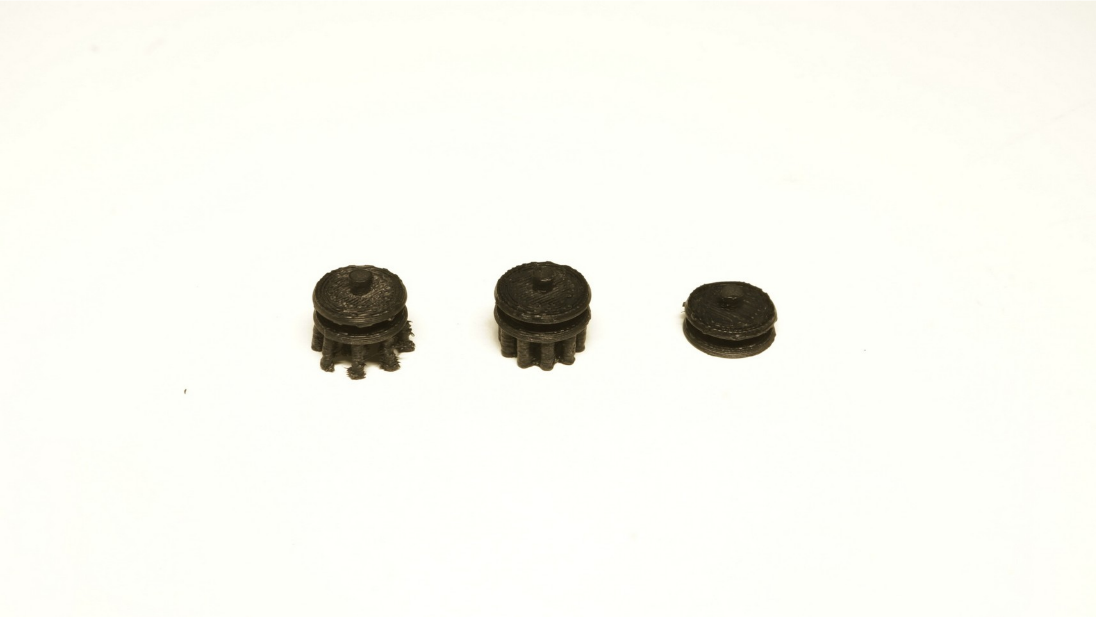


Figure 30. Headpin Spiderman (left), MultipleStuds (center) and flat (right) electrodes printed with FilaFlex conductive material and Electrically Conductive Paint

4.1.4 Comparison with Commercial Products

In this section, it was compared the performance of a custom-designed EEG headset and electrodes (Headpin Spiderman electrode printed with Conductive Filaflex with Electrically Conductive Paint as coating) against commercial products (Enobio headcap and electrodes). An OpenBCI Cyton board was utilized to record EEG signals from both systems, then signal quality was evaluated.

The results revealed that both custom-designed EEG headset and electrodes performed well, while the commercial system excelled slightly more in terms of signal quality and reduced noise level (Table 3). Impedance and resistance values fell within acceptable ranges in comparison to the original Enobio electrodes as it can be seen in tables 1 and 2. Electrode positioning was accurate and consistent across measurements when using the 3D printed headset and the custom designed electrodes offered greater comfort levels overall compared to the commercial ones.

Overall, our custom-designed EEG headset and electrodes provide an efficient and cost-effective alternative to commercial systems in terms of comfort, fit, ease of use, signal quality and overall signal reliability. These results support their suitability for EEG recordings in research or clinical settings as well as provide valuable insights for future studies involving EEG data collection.

4.2 Machine Learning Model Performance Evaluation

In this section it was evaluated the performance of machine learning models using various metrics. Both CNN models trained with public dataset and transfer learning models retrained with personal data were evaluated. Results from both approaches were compared.

Overall, the machine learning techniques employed in this study proved effective at classifying EEG signals and controlling a robot arm using motor imagery tasks. Transfer learning proved particularly valuable in improving model performance and its results provided evidence for its potential in EEG signal classification for controlling robot arms.

4.2.1 CNN Models Performance

The performance of the CNN models were evaluated using different metrics to assess their effectiveness in predicting the "motor imagery left hand" and "motor imagery right hand" classes. For evaluating the performance of every model, different metrics were used such as Cohen's Kappa Coefficient, Classification Report and Confusion Matrix (Powers, 2007, pp. 37-38). Cohen's Kappa Coefficient was computed to measure agreement between predicted and true labels. The classification report details precision, recall and F1-score for each class. Confusion Matrix was used to depict the performance of the models in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The rows of the confusion matrix represent true labels, while the columns depict predicted labels. Values on the diagonal indicate correct predictions (TP and TN), while those outside represent misclassifications (FP and FN).

4.2.1.1 HopefullNet Evaluation Metrics

Cohen's Kappa Coefficient showed that an agreement level was reached of 0.181 which indicated a low level of agreement. Table 4 summarizes the Classification Report which were calculated to evaluate the performance of the HopefullNet model after training. In terms of results for "motor imagery left hand" class results: precision of 0.50; recall of 1.00; F1-score of 0.67. In terms of "motor imagery right hand" class, precision was 0.83 with recall at 0.02 and F1 score being only 0.05 (indicating poorer performance compared to "motor imagery right hand"). Overall accuracy for model was only 0.51 which can be considered low.

Table 4. Precision, recall and F1-score for HopefullNet model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.5	1.0	0.67
Motor Imagery Right Hand	0.83	0.02	0.05

According to the confusion matrix (Table 5), 215 samples for "motor imagery left hand" class were correctly predicted while 1 sample from "motor imagery left hand" class was incorrectly misclassified into "motor imagery right hand" class by model. On the other hand, 5 samples from "motor imagery right hand" were correctly predicted while 211 samples for "motor imagery right hand" were misclassified as being "motor imagery left hand" class.

Table 5. Confusion Matrix for HopefullNet model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	215	1
Motor Imagery Right Hand	211	5

In summary, the evaluation metrics indicate that the HopefullNet model achieved very low performance in predicting the "motor imagery left hand" and "motor imagery right hand" classes, with a low level of agreement between predicted and true labels. Further improvements may be needed to enhance the model's performance for the given task.

4.2.1.2 FullyConnectedNet Evaluation Metrics

Cohen's Kappa Coefficient showed a value of 0.0 indicated no agreement between predicted and true labels. After training, the FullyConnectedNet model's Classification Report was calculated and results displayed in Table 6. For the "motor imagery left hand" class, precision was 0.50 with recall being 1.00 and F1 score being 0.67; similarly, in regard to "motor imagery right hand" class they all came back as 0.0 which indicates poor performance. Overall model accuracy was 0.50 which signifies very poor performance.

Table 6. Precision, recall and F1-score for FullyConnectedNet model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.5	1.0	0.67
Motor Imagery Right Hand	0.0	0.0	0.0

As shown in the confusion matrix (Table 7), all 216 samples from "motor imagery left hand" class were correctly predicted. All samples from "motor imagery right hand" class were misclassified as "motor imagery left hand". The model, in order to obtain the highest score possible, wrongly classified every "motor imagery right hand" samples as "motor imagery left hand".

Table 7. Confusion Matrix for FullyConnectedNet model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	216	0
Motor Imagery Right Hand	216	0

Overall, evaluation metrics reveal poor performance by the FullyConnectedNet model in predicting "motor imagery right hand" classes since the model can only correctly predict "motor imagery left hand". Results shows no agreement between predicted and true labels. Therefore, further improvements may be required in order to enhance its performance for this task.

4.2.1.3 DenseNet Evaluation Metrics

Cohen's Kappa Coefficient showed with an agreement rate of 0.268 indicating a fair agreement between predicted and true labels. The DenseNet model's Classification Report was computed after training and it is presented in Table 8. The "motor imagery left hand" class had precision of 0.65, recall of 0.59 and an F1-score of 0.62. On the "motor imagery right hand" class, these scores were 0.62 for precision, 0.68 for recall and 0.65 for F1-score with overall model accuracy standing at 0.63 (indicating moderate performance).

Table 8. Precision, recall and F1-score for DenseNet model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.65	0.59	0.62
Motor Imagery Right Hand	0.62	0.68	0.65

According to the confusion matrix (Table 9), the DenseNet model successfully classified 127 samples for "motor imagery left hand" class while misclassifying 89 of them from "motor imagery left hand" as "motor imagery right hand". 69 samples for "motor imagery right hand" were correctly predicted and 69 samples for "motor imagery right hand" were wrongly predicted as "motor imagery left hand".

Table 9. Confusion Matrix for DenseNet model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	127	89
Motor Imagery Right Hand	69	147

Evaluation metrics indicated moderate performance by DenseNet model when predicting both "motor imagery left hand" and "motor imagery right hand" classes. The model shows moderate accuracy with fair agreement between predicted and actual labels. Further optimizations may be required to improve its effectiveness for specific tasks.

4.2.1.4 SimplifiedConvNet Evaluation Metrics

Cohen's Kappa Coefficient had an agreement value of 0.754, signifying substantial agreement. To evaluate the performance of the SimplifiedConvNet model after training, Classification Report was calculated and summarized in Table 10. For the "motor imagery left hand" class these measurements were as follows: precision of 0.89, recall of 0.86 and F1-score of 0.88 respectively. On the "motor imagery right hand" class, these scores were 0.87 for precision, 0.89 for recall and 0.88 for F1-score while 0.88 overall accuracy indicated good performance by this model.

Table 10. Precision, recall and F1-score for SimplifiedConvNet model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.89	0.86	0.88
Motor Imagery Right Hand	0.87	0.89	0.88

According to the confusion matrix (Table 11), the SimplifiedConvNet model correctly predicted 186 samples belonging to "motor imagery left hand" class and 193 samples belonging to "motor imagery right hand" class. On the other hand, 30 samples for "motor imagery left hand" were misclassified as "motor imagery right hand" and 23 samples for "motor imagery right hand" were misclassified as "motor imagery left hand".

Table 11. Confusion Matrix for SimplifiedConvNet model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	186	30
Motor Imagery Right Hand	23	193

Overall, evaluation metrics demonstrated excellent performance by the SimplifiedConvNet model in accurately predicting both "motor imagery left hand" and "motor imagery right hand" classes with substantial agreement between predicted and actual labels. It seems this model has done its job effectively for this task.

4.2.1.5 TimeSeriesCNN Evaluation Metrics

Cohen's Kappa Coefficient showed with an outstanding value of 0.861 being obtained, signifying high agreement between predicted and true labels. The TimeSeriesCNN model's Classification Report is displayed in Table 12. In the "motor imagery left hand" class the results were as follows: precision 0.92, recall 0.94 and F1-score 0.93 while for "motor imagery right hand" class, precision was 0.94 with recall at 0.92 and F1 score at 0.93 respectively, all indicators of excellent model performance. Overall accuracy for the model stood at 0.93 which signifies its success in accurate prediction.

Table 12. Precision, recall and F1-score for TimeSeriesCNN model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.92	0.94	0.93
Motor Imagery Right Hand	0.94	0.92	0.93

According to the confusion matrix (Table 13), 203 samples for "motor imagery left hand" class and 199 from "motor imagery right hand" class were predicted correctly. 13 samples for "motor imagery left hand" were misclassified as "motor imagery right hand" and 17 samples for "motor imagery right hand" were wrongly labeled as "motor imagery left hand".

Table 13. Confusion Matrix for TimeSeriesCNN model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	203	13
Motor Imagery Right Hand	17	199

Evaluation metrics demonstrate outstanding performance of TimeSeriesCNN model in predicting both "motor imagery left hand" and "motor imagery right hand" classes accurately, with high agreement between predicted labels and true labels. Overall, this model appears to be performing exceptionally for classifying new data from the public dataset.

This model appears to outperform the rest of the models in terms of hand motor imagery prediction. It displays greater agreement between predicted and true labels, higher precision, recall, F1-scores for both classes, and reduced misclassifications in confusion matrix.

4.2.1.6 EfficientConvNet Evaluation Metrics

Cohen's Kappa Coefficient had an agreement value of 0.865, which indicates high level of agreement. After training, the EfficientConvNet model’s Classification Report was calculated and presented in Table 14. For the "motor imagery left hand" class, precision was 0.94, recall was 0.92 and F1-score 0.93. For "motor imagery right hand", precision was 0.92, recall was 0.94 and F1-score 0.93. These results demonstrate high accuracy when it comes to predicting both classes, and their performances are similar for both "motor imagery left hand" and "motor imagery right hand" classes.

Table 14. Precision, recall and F1-score for EfficientConvNet model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.94	0.92	0.93
Motor Imagery Right Hand	0.92	0.94	0.93

The confusion matrix (Table 15) shows that our model correctly predicted 199 samples for "motor imagery left hand" class and 204 for "motor imagery right hand" class with only 17 misclassifications for "motor imagery left hand" as "motor imagery right hand" and 12 samples from "motor imagery right hand" as "motor imagery left hand".

Table 15. Confusion Matrix for EfficientConvNet model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	199	17
Motor Imagery Right Hand	12	204

Evaluation metrics revealed an excellent performance of EfficientConvNet model in predicting both "motor imagery left hand" and "motor imagery right hand" classes. Results shows an excellent agreement between predicted and true labels. Furthermore, the model appears to perform equally well for both classes in its analysis of data sets.

4.2.1.7 LiteConvNet Evaluation Metrics

Cohen's Kappa Coefficient showed a value of 0.824 indicating a moderate level of agreement. LiteConvNet model's Classification Report was calculated and presented in Table 16. For the "motor imagery left hand" class, precision was 0.92, recall 0.91 and F1-score 0.91. For "motor imagery right hand", precision was 0.91, recall 0.92 and F1-score 0.91. These results indicate high accuracy when it comes to predicting both classes; with similar performance between "motor imagery left hand" and "motor imagery right hand" classes although performance slightly decreased from previous models.

Table 16. Precision, recall and F1-score for LiteConvNet model

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.92	0.91	0.91
Motor Imagery Right Hand	0.91	0.92	0.91

The confusion matrix (Table 17) shows that our model correctly predicted 196 samples for ""motor imagery left hand"" class while misclassifying 20 samples as "motor imagery right hand" class. On the other hand, 18 samples for "motor imagery right hand" were correctly predicted and 18 samples for "motor imagery right hand" were misclassified as "motor imagery left hand".

Table 17. Confusion Matrix for LiteConvNet model

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	196	20
Motor Imagery Right Hand	18	198

Overall, evaluation metrics revealed moderate performance from LiteConvNet model in predicting both "motor imagery left hand" and "motor imagery right hand" classes, though slightly below previous models. While misclassification rate may be slightly higher compared with other models used, accuracy still provides reasonable predictions of hand motor imagery. Further analysis and improvement may be required to improve its performance.

4.2.2 Transfer Learning

4.2.2.1 HopefullNet Evaluation Metrics

Cohen's kappa coefficient had an approximate agreement level being established as 0.085 indicating low agreement between predicted and true labels. The HopefullNet model’s Classification Report is summarized in Table 18. Results for the "motor imagery left hand" class showed 0.53 for precision, a recall of 0.87 and F1-score of 0.66. For the “motor imagery right hand” class, precision value was 0.62, recall was 0.22 and a F1-score of 0.32. These metrics indicate low accuracy in predicting both classes.

Table 18. Precision, recall and F1-score for HopefullNet model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.53	0.87	0.66
Motor Imagery Right Hand	0.62	0.22	0.32

The confusion matrix (Table 19) shows that the model correctly predicted 53 samples of the "motor imagery left hand" class and just 13 samples of the "motor imagery right hand" class, but misclassified 8 samples of the "motor imagery left hand" class as "motor imagery right hand" class, and 47 samples of the "motor imagery right hand" class as "motor imagery left hand" class.

Table 19. Confusion Matrix for LiteConvNet model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	53	8
Motor Imagery Right Hand	47	13

Overall, evaluation metrics demonstrated that HopefullNet model performance was very poor. Accuracy for motor imagery left hand and right hand classes and agreement between predicted and true labels was significantly low. Additional analysis and improvement are likely required to enhance its performance on this specific task.

4.2.2.2 FullyConnectedNet Evaluation Metrics

Cohen's kappa coefficient showed with an approximate agreement level being established as 0.0 indicating no agreement between predicted and true labels. The FullyConnectedNet model's Classification Report is summarized in Table 20. Results for the "motor imagery left hand" class showed 0.0 for precision, a recall of 0.0 and F1-score of 0.0. For the “motor imagery right hand” class, precision value was 0.5, recall was 1.0 and a F1-score of 0.66.

Table 20. Precision, recall and F1-score for FullyConnectedNet model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.0	0.0	0.0
Motor Imagery Right Hand	0.5	1.0	0.66

The confusion matrix (Table 21) shows how the model failed to predict any samples of the "motor imagery left hand" class correctly, resulting in all samples being misclassified as "motor imagery right hand" class. The model, in order to obtain the highest score possible, wrongly classified every "motor imagery left hand" samples as "motor imagery right hand".

Table 21. Confusion Matrix for FullyConnectedNet model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	0	61
Motor Imagery Right Hand	0	60

Overall, evaluation metrics showed that FullyConnectedNet model performance was very poor in predicting the "motor imagery left hand" class, while achieving perfect accuracy in predicting the "motor imagery right hand" class. Additional analysis and improvement are likely required to enhance its performance on motor imagery classification.

4.2.2.3 DenseNet Evaluation Metrics

Cohen's kappa coefficient obtained an approximate agreement level being established as 0.052 indicating a slight agreement between predicted and true labels. Table 22 provides a summary of the model’s Classification Report. Results for the "motor imagery left hand" class showed 0.52 for precision, a recall of 0.87 and F1-score of

0.65. For the “motor imagery right hand” class, precision value was 0.58, recall was 0.18 and a F1-score of 0.28. These metrics indicate that the model achieved relatively higher accuracy in predicting the "motor imagery left hand" class compared to the "motor imagery right hand" class.

Table 22. Precision, recall and F1-score for DenseNet model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.52	0.87	0.65
Motor Imagery Right Hand	0.58	0.18	0.28

The confusion matrix (Table 23) shows how the model correctly predicted 53 samples of the "motor imagery left hand" class and 11 samples of the "motor imagery right hand" class, but misclassified 8 samples of the "motor imagery left hand" class as "motor imagery right hand" class, and 49 samples of the "motor imagery right hand" class as "motor imagery left hand" class.

Table 23. Confusion Matrix for DenseNet model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	53	8
Motor Imagery Right Hand	49	11

Overall, evaluation metrics demonstrated that DenseNet model performance was relatively better with almost no difference in predicting the "motor imagery left hand" class compared to the "motor imagery right hand" class. Additional analysis and improvement are likely required to enhance the performance of the model on motor imagery classification.

4.2.2.4 SimplifiedConvNet Evaluation Metrics

Cohen's kappa coefficient had an approximate agreement level being established as 0.718 indicating a substantial agreement between predicted and true labels. The SimplifiedConvNet model's Classification Report is presented in Table 24. Results for the "motor imagery left hand" class showed 0.8 for precision, a recall of 0.97 and F1-score of 0.87. For the “motor imagery right hand” class, precision value was 0.96, recall was 0.75 and a F1-score of 0.84. These metrics indicate that the model achieved high accuracy in predicting both classes, with relatively higher accuracy for the "motor imagery left hand" class.

Table 24. Precision, recall and F1-score for SimplifiedConvNet model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.8	0.97	0.87
Motor Imagery Right Hand	0.96	0.75	0.84

The confusion matrix (Table 25) shows that the model correctly predicted 59 samples of the "motor imagery left hand" class and 45 samples of the "motor imagery right hand" class, but misclassified 2 samples of the "motor imagery left hand" class as "motor imagery right hand" class, and 15 samples of the "motor imagery right hand" class as "motor imagery left hand" class.

Table 25. Confusion Matrix for SimplifiedConvNet model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	59	2
Motor Imagery Right Hand	15	45

Overall, evaluation metrics demonstrated that SimplifiedConvNet model achieved high accuracy in predicting both "motor imagery left hand" and "motor imagery right hand" classes. The model showed a substantial agreement between predicted and true labels, with relatively higher accuracy for the "motor imagery left hand" class. However, further analysis and fine-tuning may be required to optimize the model's performance and minimize misclassifications for both classes.

4.2.2.5 TimeSeriesCNN Evaluation Metrics

Cohen's kappa coefficient obtained an approximate agreement level being established as 0.619 indicating a substantial agreement between predicted and true labels. The TimeSeriesCNN model's Classification Report is summarized in Table 26. Results for the "motor imagery left hand" class showed 0.8 for precision, a recall of 0.84 and F1-score of 0.82. For the “motor imagery right hand” class, precision value was 0.82, recall was 0.78 and a F1-score of 0.8. These metrics indicate that the model achieved high accuracy in predicting both classes, with relatively balanced accuracy for both "motor imagery left hand" and "motor imagery right hand" classes.

Table 26. Precision, recall and F1-score for TimeSeriesCNN model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.8	0.84	0.82
Motor Imagery Right Hand	0.82	0.78	0.8

The confusion matrix (Table 27) shows how the model correctly predicted 51 samples of the "motor imagery left hand" class and 47 samples of the "motor imagery right hand" class, but misclassified 10 samples of the "motor imagery left hand" class as "motor imagery right hand" class, and 13 samples of the "motor imagery right hand" class as "motor imagery left hand" class.

Table 27. Confusion Matrix for TimeSeriesCNN model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	51	10
Motor Imagery Right Hand	13	47

Overall, evaluation metrics demonstrated that TimeSeriesCNN model achieved high accuracy in predicting both "motor imagery left hand" and "motor imagery right hand" classes, with relatively balanced accuracy for both classes. The model showed a substantial agreement between predicted and true labels, indicating its effectiveness in classification tasks. However, further analysis and fine-tuning may be required to optimize the model's performance and minimize misclassifications for both classes.

4.2.2.6 EfficientConvNet Evaluation Metrics

Cohen's kappa coefficient obtained an approximate agreement level being established as 0.669 indicating a substantial agreement between predicted and true labels. The EfficientConvNet model’s Classification Report is summarized in Table 28. Results for the "motor imagery left hand" class showed 0.9 for precision, a recall of 0.75 and F1-score of 0.82. For the “motor imagery right hand” class, precision value was 0.79, recall was 0.92 and a F1-score of 0.85. These metrics indicate that the model achieved high precision and recall for both "motor imagery left hand" and "motor imagery right hand" classes, with relatively balanced accuracy for both classes.

Table 28. Precision, recall and F1-score for EfficientConvNet model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.9	0.75	0.82
Motor Imagery Right Hand	0.79	0.92	0.85

The confusion matrix (Table 29) shows how the model correctly predicted 46 samples of the "motor imagery left hand" class and 55 samples of the "motor imagery right hand" class, but misclassified 15 samples of the "motor imagery left hand" class as "motor imagery right hand" class, and 5 samples of the "motor imagery right hand" class as "motor imagery left hand" class.

Table 29. Confusion Matrix for EfficientConvNet model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	46	15
Motor Imagery Right Hand	5	55

Overall, evaluation metrics demonstrated that EfficientConvNet model achieved high accuracy in predicting both "motor imagery left hand" and "motor imagery right hand" classes, with relatively balanced accuracy for both classes. The model showed a substantial agreement between predicted and true labels, indicating its effectiveness in classification tasks. However, further analysis and fine-tuning may be required to optimize the model's performance and minimize misclassifications for both classes.

4.2.2.7 LiteConvNet Evaluation Metrics

Cohen's kappa coefficient obtained an approximate agreement level being established as 0.636 indicating a substantial agreement between predicted and true labels. The LiteConvNet model's Classification Report is summarized in Table 30. Results for the "motor imagery left hand" class showed 0.88 for precision, a recall of 0.74 and F1-score of 0.8. For the “motor imagery right hand” class, precision value was 0.77, recall was 0.9 and a F1-score of 0.83. These metrics indicate that the model achieved high precision and recall for both "motor imagery left hand" and "motor imagery right hand" classes, with relatively balanced accuracy for both classes.

Table 30. Precision, recall and F1-score for LiteConvNet model using Transfer Learning

True/Prediction	Precision	Recall	F1-score
Motor Imagery Left Hand	0.88	0.74	0.8
Motor Imagery Right Hand	0.77	0.9	0.83

The confusion matrix (Table 31) shows how the model correctly predicted 45 samples of the "motor imagery left hand" class and 54 samples of the "motor imagery right hand" class, but misclassified 16 samples of the "motor imagery left hand" class as "motor imagery right hand" class, and 6 samples of the "motor imagery right hand" class as "motor imagery left hand" class.

Table 31. Confusion Matrix for LiteConvNet model using Transfer Learning

True/Prediction	Motor Imagery Left Hand	Motor Imagery Right Hand
Motor Imagery Left Hand	45	16
Motor Imagery Right Hand	6	54

Overall, evaluation metrics demonstrated that LiteConvNet model achieved high accuracy in predicting both "motor imagery left hand" and "motor imagery right hand" classes, with relatively balanced accuracy for both classes. The model showed a substantial agreement between predicted and true labels, indicating its effectiveness in classification tasks. However, further analysis and fine-tuning may be required to optimize the model's performance and minimize misclassifications for both classes.

4.3 Robot Control using EEG

EEG signals were utilized to control an Elfin5 robot arm using ROS and MoveIt package. Accuracy and efficacy were assessed through movement task evaluation (Roy et al., 2016, pp. 149-151).

Results demonstrated that using EEG data collected during own measurements, it was possible to successfully control robot arm movements using them as commands for movement control. Movement was smooth and precise demonstrating the system's capability of accurately deciphering EEG signals into meaningful movement commands.

It was obtained a high degree of satisfaction with the system, particularly its ease of use and comprehensive control over robot arm movements. Overall, these results indicate that using EEG signals for robotic control could be an efficient and reliable technique in numerous applications.

4.3.1 Movement Task Execution

The movement task execution using the trained TimeSeriesCNN model and the Elfin5 Robot Arm in simulation was successful. The user's motor imagery of left and right hand was accurately classified based on the EEG data acquired and preprocessed. Furthermore, the robot arm executed the corresponding joint space trajectory with its end effector for picking up an object from left or right side before transporting it towards center using cartesian space trajectory. Then, the robot arm returned back to its original position waiting for the next prediction. This demonstrated the potential of using EEG-based motor imagery classification for controlling a robot arm in a simulated environment.

4.3.2 Accuracy and Robustness

Accuracy and robustness of robot control system were assessed based on the performance of the trained TimeSeriesCNN model in classifying motor imagery from EEG data. Motor imagery classification accuracy was evaluated using metrics such as accuracy, precision, recall and F1-score. Robustness was measured by testing its performance under various conditions such as changing EEG data quality (Hu & Wang, 2017, pp. 3-11) and different environments.

Results demonstrated that the trained TimeSeriesCNN model achieved relatively high accuracy when classifying motor imagery from EEG data, with an average accuracy rate of over 62.50%. This lower accuracy was obtained when testing the model with EEG data acquired in different environments. Furthermore, precision recall and F1-score were relatively good, showing its ability to accurately classify both left and right hand motor imagery in a controlled environment. Furthermore, this system demonstrated robustness over different differing EEG data quality levels but not in different environments.

Overall, our results indicate that the robot control system using EEG-based motor imagery classification and the Elfin5 Robot Arm in simulation is both accurate and robust enough for real world use, especially when a controlled environment can be guaranteed, suggesting its potential use in controlling robot arms using EEG signals. Further discussion of the results will be presented in the following sections.

5 DISCUSSION

5.1 Interpretation of Results

Interpreting the results requires a thorough examination of the study details and explaining their significance and implications. In this section, it was interpreted the outcomes of the investigation on motor imagery-based EEG signal classification and robot control.

This study revealed that the custom-designed EEG headset and electrode configuration performed well compared to commercial products. The optimized positioning of electrodes resulted in high-quality EEG signals, and the materials and coatings used exhibited low resistance and impedance. These findings suggest that our EEG headset and electrode setup could be suitable not only for motor imagery-based EEG signal classification but also for other EEG-based applications.

The evaluation of Machine Learning Model Performance demonstrated that CNN models that were trained achieved excellent accuracy when measuring EEG signals originating in same environment, yet substantially decreased with noisy environments or when measurements were taken in a different environment. Transfer learning also improved CNN model performance when our dataset was limited, specially by enhancing generalization with other public datasets. These outcomes indicate that CNN models hold promise as a solution for motor imagery-based EEG signal classification.

The results of our experiments utilizing EEG signals revealed that Elfin5 robot arm could accurately and reliably execute movement tasks designed with ROS MoveIt package, though success rates decreased in noisy environments since training measurements had been collected under non-disruptive conditions.

Overall, the study demonstrated the feasibility and potential of motor imagery-based EEG signal classification and robot control. The outcomes of the investigation can help advance EEG technologies and their applications across various fields, such as robotics.

5.2 Limitations and future work

This research had some limitations, such as its small sample size which may limit generalizability of results and using only one EEG headset which may limit accuracy due to variances in electrode placement and materials used.

Future studies should incorporate larger sample sizes to increase generalizability of results and test various EEG headsets and electrode materials to find which configuration gives the most precise and reliable readings.

One major issue faced in the thesis is that the model has limited robustness, as it was only able to perform excellently under controlled conditions. The performance of the model may be compromised when exposed to real-world variability, noise, and unpredictable conditions. In addition, the model should be tested and refined using real-world data to address the challenges and challenges encountered outside of the controlled environment.

Additionally, this study could be expanded to incorporate more complex movement tasks and use of multiple robot arms for practical applications. Machine learning algorithms and models could also be explored to increase accuracy and speed of EEG-based control system, aiming for real-time classification.

Finally, integrating other physiological signals such as Electromyography (EMG) may further improve accuracy and dependability of the system.

6 CONCLUSION

In conclusion, this thesis presented a novel method for robot control using EEG signals. This included designing and fabricating a custom EEG headset and electrodes, acquiring and preprocessing EEG signals for classification by machine learning models, connecting the results to an automated robotic arm using ROS MoveIt package and connecting EEG signals directly as control for automation purposes.

Results demonstrated that the custom EEG headset and electrodes configuration achieved satisfactory impedance and noise levels comparable to commercial EEG devices. Machine learning models developed for this thesis showed high accuracy in classifying EEG signals associated with motor imagery tasks, with transfer learning approach using pretrained deep neural network being the top performer in terms of consistency and generalization. Finally, the EEG-based robot control system successfully executed specified movement tasks with sufficient precision and robustness.

However, this thesis also highlighted some limitations to its proposed approach, including a limited variety of movement tasks considered, the small sample size for participants in the experiment and lack of different EEG headsets and electrode materials for testing. Future work could address these shortcomings by including more movement tasks, involving a larger and more diverse sample group and employing multiple headsets and materials for testing purposes.

Overall, this thesis offers a proof-of-concept for using EEG signals to control a robotic arm, potentially revolutionizing the field of robotics.

REFERENCES

A Motor-Imagery BCI System Based on Deep Learning Networks and Its Applications - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Locations-C3-C4-and-Cz-are-used-in-the-10-20-system_fig1_328351824 (accessed April 20, 2023)

Abdeltawab, A., & Ahmad, A. (2020, November 9). Classification of Motor Imagery EEG Signals Using Machine Learning. Paper presented at the 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), Shah Alam, Malaysia. https://doi.org/10.1109/ICSET51301.2020.9265364

Acharya, J. N., & Acharya, V. J. (2019). Overview of EEG montages and principles of localization. Journal of Clinical Neurophysiology, 36(5), 325-329. https://doi.org/10.1097/WNP.0000000000000538

Amid, E., Anil, R., Kotłowski, W., & Warmuth, M. K. (2023). Learning from Randomly Initialized Neural Network Features. arXiv preprint arXiv:arXiv:2207.08148. https://doi.org/10.48550/arXiv.2202.06438

An, Y., Lam, H.K., & Ling, S.H. (2023). Multi-classification for EEG motor imagery signals using data evaluation-based auto-selected regularized FBCSP and convolutional neural network. Neural Computing and Applications. https://doi.org/10.1007/s00521-023-08336-z

Ang, K. K., & Guan, C. (2013). Brain-Computer Interface in Stroke Rehabilitation. Journal of Computing Science and Engineering, 7(2), 139-146. https://doi.org/10.5626/JCSE.2013.7.2.139

Azlan, W. A. W., & Low, Y. F. (2015). Feature extraction of electroencephalogram (EEG) signal - A review. In 2014 IEEE Conference on Biomedical Engineering and Sciences. Miri, Sarawak, Malaysia: IEEE, 801-806. https://doi.org/10.1109/IECBES.2014.7047620

Blum, S., Hölle, D., Bleichner, M. G., & Debener, S. (2021). Pocketable Labs for Everyone: Synchronized Multi-Sensor Data Streaming and Recording on Smartphones with the Lab Streaming Layer. Sensors, 21(23), 8135. https://doi.org/10.3390/s21238135

Brunner, C., Leeb, R., Müller-Putz, G. R., Schlögl, A., & Pfurtscheller, G. (2008). BCI Competition 2008 – Graz data set A. Institute for Knowledge Discovery, Graz University of Technology, Austria; Institute for Human-Computer Interfaces, Graz University of Technology, Austria. https://lampz.tugraz.at/~bci/database/001-2014/description.pdf

Cui, J., Yuan, L., Wang, Z., Li, R., & Jiang, T. (2022). Towards Best Practice of Interpreting Deep Learning Models for EEG-based Brain Computer Interfaces. arXiv preprint arXiv:2202.06948. https://doi.org/10.48550/arXiv.2202.06948

De Cheveigné, A., & Nelken, I. (2019). Filters: When, why, and how (not) to use them. Neuron, 102(2), 280-293. https://doi.org/10.1016/j.neuron.2019.02.039

Desai, J. (2014). Electroencephalography(EEG) Data Collection and Processing through Machine Learning. [Master's thesis, University of Arkansas]. Graduate Theses and Dissertations. https://scholarworks.uark.edu/etd/2160

Douibi, K., Le Bars, S., Lemontey, A., Nag, L., Balp, R., & Breda, G. (2021). Toward EEG-Based BCI Applications for Industry 4.0: Challenges and Possible Applications. Frontiers in Human Neuroscience, 15, 705064. https://doi.org/10.3389/fnhum.2021.705064

Fiedler, P., Brodkorb, S., Fonseca, C., & Vaz, F. (2009). Novel TiN-based dry EEG electrodes: Influence of electrode shape and number on contact impedance and signal quality. In IFMBE Proceedings, 29, 418-421. Springer. https://doi.org/10.1007/978-3-642-13039-7_105

Guillot, A., Di Rienzo, F., MacIntyre, T., Moran, A., & Collet, C. (2012). Imagining is not doing but involves specific motor commands: A review of experimental data related to motor inhibition. Frontiers in human neuroscience, 6, 247. https://doi.org/10.3389/fnhum.2012.00247

Hans Robot. (2023). xElfin-2023. Hans Robot. https://www.hansrobot.net/media/upload/product/elfin/xElfin-2023.2.png.pagespeed.ic.awpGSTe_5b.png (accessed April 21, 2023).

Herath, H. M. K. K. M. B., & deMel, W. R. (2021). Controlling an anatomical robot hand using the brain-computer interface based on motor imagery. Advances in Human-Computer Interaction, 2021, Article ID 5515759. https://doi.org/10.1155/2021/5515759

Hernandez-Mendez, S., Maldonado-Mendez, C., Marin-Hernandez, A., Rios-Figueroa, H. V., Vazquez-Leal, H., & Palacios-Hernandez, E. R. (2017). Design and implementation of a robotic arm using ROS and MoveIt!. In 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), 1-6. IEEE. https://doi.org/10.1109/ROPEC.2017.8261666

Hortal, E., Planelles, D., Costa-Garcia, A., & Iáñez, E. (2015). SVM-based Brain–Machine Interface for controlling a robot arm through four mental tasks. Neurocomputing, 151(1), 116-121. https://doi.org/10.1016/j.neucom.2014.09.078

Hu, J., & Wang, P. (2017). Noise Robustness Analysis of Performance for EEG-Based Driver Fatigue Detection Using Different Entropy Feature Sets. Entropy, 19(8), 385. https://doi.org/10.3390/e19080385

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4700-4708. https://doi.org/10.48550/arXiv.1608.06993

Jamil, N., Belkacem, A. N., Ouhbi, S., & Lakas, A. (2021). Noninvasive electroencephalography equipment for assistive, adaptive, and rehabilitative brain–computer interfaces: A systematic literature review. Sensors, 21(14), 4754. https://doi.org/10.3390/s21144754

Jeong, J.-H., Shim, K.-H., Kim, D.-J., & Lee, S.-W. (2020). Brain-Controlled Robotic Arm System Based on Multi-Directional CNN-BiLSTM Network Using EEG Signals. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 28(5), 1226-1235. https://doi.org/10.1109/TNSRE.2020.2981659

Jha, R., Bhattacharjee, V., & Mustafi, A. (2022). Transfer Learning with Feature Extraction Modules for Improved Classifier Performance on Medical Image Data. Volume 2022, Article ID 4983174. https://doi.org/10.1155/2022/4983174

Jia, H., Feng, F., Caiafa, C. F., Duan, F., Zhang, Y., Sun, Z., & Solé-Casals, J. (2022). Towards a multi-class classification of upper limb movements. Journal of Neural Engineering, 19(1), 016031. https://arxiv.org/pdf/2201.12157.pdf

Jiang, X., Bian, G.-B., & Tian, Z. (2019). Removal of artifacts from EEG signals: A review. Sensors (Basel), 19(5), 987. https://doi.org/10.3390/s19050987

Ju, C., Gao, D., Mane, R., Tan, B., Liu, Y., & Guan, C. (2020). Federated Transfer Learning for EEG Signal Classification. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 3040-3045. https://doi.org/10.1109/EMBC44109.2020.9175344

Kang, J. -S., Kavuri, S., & Lee, M. ICA-Evolution Based Data Augmentation with Ensemble Deep Neural Networks Using Time and Frequency Kernels for Emotion Recognition from EEG-Data, in IEEE Transactions on Affective Computing, vol. 13, no. 2, 616-627, 1 April-June 2022, https://doi.org/10.1109/TAFFC.2019.2942587

Kim, C., Sun, J., Liu, D., & Wang, Q. (2018). An effective feature extraction method by power spectral density of EEG signal for 2-class motor imagery-based BCI. Medical & Biological Engineering & Computing, 56(2), 1-14. https://doi.org/10.1007/s11517-017-1761-4

Kim, Y., Ryu, J., Kim, K. K., Took, C. C., Mandic, D. P., & Park, C. (2016). Motor Imagery Classification Using Mu and Beta Rhythms of EEG with Strong Uncorrelating Transform Based Complex Common Spatial Patterns. Computational Intelligence and Neuroscience, 2016, 1489692. https://doi.org/10.1155/2016/1489692

Koctúrová, M., & Juhár, J. (2019). Comparison of dry electrodes for mobile EEG system. Conference on Theory and Materials Science. Technical University of Košice. https://ceur-ws.org/Vol-2473/paper36.pdf

Korovesis, N., Kandris, D., Koulouras, G., & Alexandridis, A. (2019). Robot motion control via an EEG-based brain–computer interface by using neural networks and alpha brainwaves. Electronics, 8(12), 1387. https://doi.org/10.3390/electronics8121387

Krachunov, S., & Casson, A. (2016). 3D Printed Dry EEG Electrodes. Sensors (Basel, Switzerland), 16(10), 1637. https://doi.org/10.3390/s16101635

Kulkarni, K.K., & Bhosale, D.S. (2018). EEG-Based Brain-Controlled Mobile Robots. International Journal of Electrical and Electronics Research, 6(2), 62-66. https://www.researchpublish.com/upload/book/EEG-Based%20Brain-Controlled-5700.pdf

Lacko, D., Vleugels, J., Fransen, E., Huysmans, T., De Bruyne, G., Van Hulle, M. M., Sijbers, J., & Verwulgen, S. (2016). Ergonomic design of an EEG headset using 3D anthropometry. iMinds-Vision Lab, University of Antwerp, Antwerp, Belgium. https://doi.org/10.1016/j.apergo.2016.06.002

Lan, T. (2011). Feature extraction, feature selection and dimensionality reduction techniques for Brain Computer Interfaces (Doctoral dissertation, Oregon Health & Science University). https://scholararchive.ohsu.edu/downloads/fn106z027?locale=pt-BR

Lawhern, V.J., Solon, A.J., Waytowich, N.R., Gordon, S.M., Hung, C.P., & Lance, B.J. (2018). EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. ArXiv. https://doi.org/10.48550/arXiv.1611.08024

Lin, J. S., & Shih, R. (2018). A Motor-Imagery BCI System Based on Deep Learning Networks and Its Applications. In Evolving BCI Therapy - Engaging Brain State Dynamics, 65-84. IntechOpen. https://doi.org/10.5772/intechopen.75009

Liu, L., Shi, C., & Wu, X. Low Quality Samples Detection in Motor Imagery EEG Data by Combining Independent Component Analysis and Confident Learning, in 2022 21st International Symposium on Communications and Information Technologies (ISCIT), Xi'an, China, 2022, 7-12. https://doi.org/10.1109/ISCIT55906.2022.9931282

Lun, X., Yu, Z., Chen, T., Wang, F., & Hou, Y. (2020). A simplified CNN classification method for MI-EEG via the electrode pairs signals. Frontiers in Human Neuroscience, 14, 338. https://doi.org/10.3389/fnhum.2020.00338

Mao, W-L., Fathurrahman, H. I. K., Lee, Y., & Chang, T. W. (2020). EEG dataset classification using CNN method. Journal of Physics: Conference Series, 1456, 012017. https://doi.org/10.1088/1742-6596/1456/1/012017

Mc1ntyre, J. (2022, October 26th). DeepLearningProject. GitHub Repository. Retrieved March 11, 2023, from https://github.com/James-Mc1ntyre/DeepLearningProject

Mitocaru, A., Poboroniuc, M. S., Irimia, D., & Baciu, A. (2021). Comparison between two Brain Computer Interface systems aiming to control a mobile robot. In 2021 International Conference on Electromechanical and Energy Systems (SIELMEN), 1-6. IEEE. https://doi.org/10.1109/SIELMEN53755.2021.9600389

Montoya-Martínez, J., Vanthornhout, J., Bertrand, A., & Francart. T. (2021). Effect of number and placement of EEG electrodes on measurement of neural tracking of speech. PLoS One, 16(2), e0246769. https://doi.org/10.1371/journal.pone.0246769

Morales S. & Bowers M. Time-frequency analysis methods and their application in developmental EEG data. Developmental Cognitive Neuroscience. 2022;54:101067. https://doi.org/10.1016/J.DCN.2022.101067

Motdhare, S. S., & Mathur, G. (2022). Eye Blink Artifact Removal from Cognitive EEG Data using ICA (Independent Component Analysis). Mathematical Statistician and Engineering Applications, 71(1), 86-93. https://doi.org/10.17762/msea.v71i1.45

Muhammad, R., Ali, A., & Anwar, M. A. (2023). Design and Development of Low-cost Wearable Electroencephalograms (EEG) Headset. Intelligent Automation and Soft Computing, 35(5), 2821–2835. https://doi.org/10.32604/iasc.2023.026279

Nishimura, K., Ishikawa, T., Sasaki, H., & Kato, S. (2021, August). RAPLET: Demystifying Publish/Subscribe Latency for ROS Applications. In 2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 1-11. IEEE. https://doi.org/10.1109/RTCSA52859.2021.00013

Onose, G., Grozea, C., Anghelescu, A., Daia, C., Sinescu, C. J., Ciurea, A. V., Spircu, T., Mirea, A., Andone, I., Spaˆnu, A., Popescu, C., Miha˘escu, A.-S., Fazli, S., Dano´ czy, M., & Popescu, F. (2012). On the feasibility of using motor imagery EEG-based brain–computer interface in chronic tetraplegics for assistive robotic arm control: a clinical test and long-term post-trial follow-up. Spinal Cord, 50(8), 599–608. https://doi.org/10.1038/sc.2012.14

Padfield, N., Zabalza, J., Zhao, H., Masero, V., & Ren, J. (2019). EEG-Based Brain-Computer Interfaces Using Motor-Imagery: Techniques and Challenges. Sensors, 19(6), 1421. https://doi.org/10.3390/s19061423

Parvan, M., Rikhtehgar Ghiasi, A., Yousefi Rezaii, T., & Farzamnia, A. (2019). Transfer Learning based Motor Imagery Classification using Convolutional Neural Networks. In 2019 27th Iranian Conference on Electrical Engineering (ICEE), 1-6. IEEE. https://doi.org/10.1109/IranianCEE.2019.8786636

Pinegger, A., Wriessnegger, S. C., Faller, J., & Müller-Putz, G. R. (2016). Evaluation of different EEG acquisition systems concerning their suitability for building a brain–computer interface: Case studies. Frontiers in neuroscience, 10, 441. https://doi.org/10.3389/fnins.2016.00441

Powers, D. M. W. (2007). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Technical Report SIE-07-001. School of Informatics and Engineering, Flinders University. https://doi.org/10.48550/arXiv.2010.16061

Procházka, A., Mudrová, M., Vyšata, O., & Hava, R. (2010). Multi-channel EEG signal segmentation and feature extraction. In Proceedings of the 14th International Conference on Intelligent Engineering Systems (INES), 95-99. IEEE. https://doi.org/10.1109/INES.2010.5483824

Puce, A., & Hämäläinen, M. S. (2017). A Review of Issues Related to Data Acquisition and Analysis in EEG/MEG Studies. Brain Sciences, 7(6), 58. https://doi.org/10.3390/brainsci7060058

Radüntz, T. (2018). Signal Quality Evaluation of Emerging EEG Devices. Frontiers in Physiology, 9, 98. https://doi.org/10.3389/fphys.2018.00098

Rahman, S. M. M., Mattila, H., & Virkki, J. (2022). Impedance evaluation of textile electrodes for EEG measurements. Textile Research Journal, 93, 7-8. https://doi.org/10.1177/00405175221135131

Rak, R., Kołodziej, M., & Majkowski, A. (2012). Brain-Computer Interface as measurement and control system. International Journal of Electronics and Telecommunications, 58(3), 275-282. https://doi.org/10.2478/v10178-012-0037-4

Robbins, K. A., Touryan, J., Mullen, T., Kothe, C., & Bigdely-Shamlo, N. (2020). How Sensitive are EEG Results to Preprocessing Methods: A Benchmarking Study. bioRxiv. https://doi.org/10.1101/2020.01.20.913327

Roy, R., Mahadevappa, M., & Kumar, C. S. (2016). Trajectory path planning of EEG controlled robotic arm using GA. Procedia Computer Science, 84, 147-151. https://doi.org/10.1016/j.procs.2016.04.080

Saibene, A., Corchs, S., Caglioni, M., & Gasparini, F. (2022). The evolution of AI approaches for motor imagery EEG-based BCIs. University of Milano-Bicocca. NeuroMI, Milan Center for Neuroscience. University of Insubria. https://doi.org/10.48550/arXiv.2210.06290

Singh, A. K., & Krishnan, S. (2023). Trends in EEG signal feature extraction applications. Frontiers in Artificial Intelligence, 5, 1072801. https://doi.org/10.3389/frai.2022.1072801

Sohal, H., Karki, B., Sharma, A., & Mohamadou, Y. (2018). Design of Impedance Measurement Module for an EEG and EIT Integrated System. International Journal of Control and Automation, 11(9), 97-108. https://doi.org/10.14257/ijca.2018.11.9.10

Tibrewal, N., Leeuwis, N., & Alimardani, M. (2022). Classification of motor imagery EEG using deep learning increases performance in inefficient BCI users. PLoS ONE, 17(7), e0268880. https://doi.org/10.1371/journal.pone.0268880

Trujillo, L. T., Stanfield-Wiswell, C. T., & Vela, R. D. (2017). The Effect of Electroencephalogram (EEG) Reference Choice on Information-Theoretic Measures of the Complexity and Integration of EEG Signals. Frontiers in Neuroscience, 11, 425. https://doi.org/10.3389/fnins.2017.00425

Tsuchimoto, S., Shibusawa, S., Iwama, S., Hayashi, M., et al. (2021). Use of common average reference and large-Laplacian spatial-filters enhances EEG signal-to-noise ratios in intrinsic sensorimotor activity. Journal of Neuroscience Methods, 353, 109089. https://doi.org/10.1016/j.jneumeth.2021.109089

Velcescu, A., Lindley, A., Cursio, C., Loh, A., & Rogers, J. (2019). Flexible 3D-printed EEG electrodes. Sensors (Switzerland), 19(7), 1676. https://doi.org/10.3390/s19071650

Värbu, K., Muhammad, N., & Muhammad, Y. (2022). Review: Past, present, and future of EEG-based BCI applications. Sensors, 22(9), 3331. https://doi.org/10.3390/s22093331

Wang, K., Wang, Z., Guo, Y., He, F., Qi, H., Xu, M., & Ming, D. An EEG study on hand force imagery for brain-computer interfaces, in 2017 8th International IEEE/EMBS Conference on Neural Engineering (NER), Shanghai, China, 2017, 236-239, https://doi.org/10.1109/NER.2017.8008439

Weber, D., Hertweck, S., Alwanni, H., Fiederer, L. D. J., Wang, X., Unruh, F., Fischbach, M., Latoschik, M. E., & Ball, T. (2021). A Structured Approach to Test the Signal Quality of Electroencephalography Measurements During Use of Head-Mounted Displays for Virtual Reality Applications. Frontiers in Neuroscience, 15, 733673. https://doi.org/10.3389/fnins.2021.733673

Wu, D., Jiang, X., & Peng, R. (2022). Transfer learning for motor imagery based brain-computer interfaces: A tutorial. Neural Networks, 153, 235-253. https://doi.org/10.1016/j.neunet.2022.06.008

Yahya, N., Musa, H., Ong, Z. Y., & Elamvazuthi, I. (2019). Classification of Motor Functions from Electroencephalogram (EEG) Signals Based on an Integrated Method Comprised of Common Spatial Pattern and Wavelet Transform Framework. Sensors (Basel), 19(22), 4878. https://doi.org/10.3390/s19224878

Zhang, K., Robinson, N., Lee, S.-W., & Guan, C. (2021). Adaptive transfer learning for EEG motor imagery classification with deep Convolutional Neural Network. Neural Networks, 136, 1-10. https://doi.org/10.1016/j.neunet.2020.12.013

Zhang, S., McCane, B., Neo, P. S.-H., & McNaughton, N. (2020). Removing mains power artefacts from EEG – a novel template-based method. bioRxiv. https://doi.org/10.1101/2020.01.27.911586

Zhang, W., Deng, L., Zhang, L., & Wu, D. (2023). A Survey on Negative Transfer. IEEE/CAA Journal of Automatica Sinica, 2022, 1-25. https://doi.org/10.1109/JAS.2022.106004

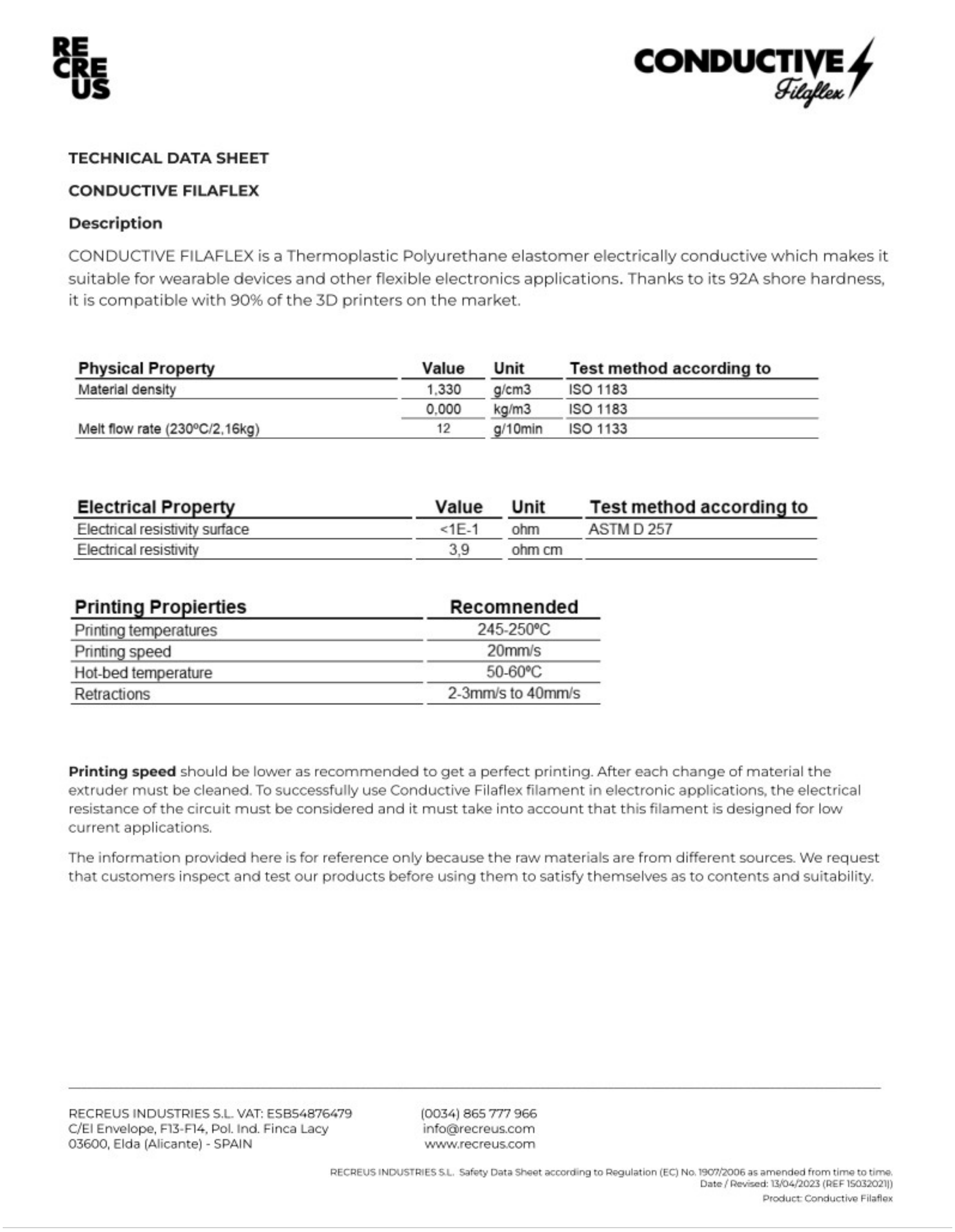
Zhang, Y., Chen, J., Tan, J. H., Chen, Y., Chen, Y., Li, D., Yang, L., Su, J., Huang, X., & Che, W. (2020). An Investigation of Deep Learning Models for EEG-Based Emotion Recognition. Frontiers in Neuroscience, 14, 622759. https://doi.org/10.3389/fnins.2020.622759

Zhao, W. (2017). Research on the deep learning of the small sample data based on transfer learning. In GREEN ENERGY AND SUSTAINABLE DEVELOPMENT I: Proceedings of the International Conference on Green Energy and Sustainable Development (GESD 2017) (Vol. 1864(1), p. 020018). AIP Publishing. https://doi.org/10.1063/1.4992835

Zhu, K., Jiao, A., & Liu, X. (2021). Design of a Low Cost EEG Headset for Educational Research. In ASEE Annual Conference and Exposition, Conference Proceedings. https://doi.org/10.18260/1-2--36917

APPENDIX 1

Conductive FilaFlex filament technical data sheet



TECHNICAL DATA SHEET

CONDUCTIVE FILAFLEX

Description

CONDUCTIVE FILAFLEX is a Thermoplastic Polyurethane elastomer electrically conductive which makes it suitable for wearable devices and other flexible electronics applications. Thanks to its 92A shore hardness, it is compatible with 90% of the 3D printers on the market.

Physical Property	Value	Unit	Test method according to
Material density	1,330	g/cm3	ISO 1183
	0,000	kg/m3	ISO 1183
Melt flow rate (230ºC/2,16kg)	12	g/10min	ISO 1133

Electrical Property	Value	Unit	Test method according to
Electrical resistivity surface	<1E-1	ohm	ASTM D 257
Electrical resistivity	3,9	ohm cm	

Printing Propierties	Recomnended
Printing temperatures	245-250ºC
Printing speed	20mm/s
Hot-bed temperature	50-60ºC
Retractions	2-3mm/s to 40mm/s

Printing speed should be lower as recommended to get a perfect printing. After each change of material the extruder must be cleaned. To successfully use Conductive Filaflex filament in electronic applications, the electrical resistance of the circuit must be considered and it must take into account that this filament is designed for low current applications.

The information provided here is for reference only because the raw materials are from different sources. We request that customers inspect and test our products before using them to satisfy themselves as to contents and suitability.

RECREUS INDUSTRIES S.L. VAT: ESB54876479
C/El Envelope, F13-F14, Pol. Ind. Finca Lacy
03600, Elda (Alicante) - SPAIN

(0034) 865 777 966
info@recreus.com
www.recreus.com

RECREUS INDUSTRIES S.L. Safety Data Sheet according to Regulation (EC) No. 1907/2006 as amended from time to time.
Date / Revised: 13/04/2023 (REF 15032021))
Product: Conductive Filaflex

Recreus. Conductive FilaFlex technical data sheet. (2023). Retrieved March 11, 2023, from:
https://drive.google.com/drive/u/1/folders/1pmSdVfz1SwgtDzKhSK_rvKH6qxcYbYtq

APPENDIX 2

Eryone TPU filament data sheet (Part 1)

Specification of TPU

Writer： **Proofreader：** **Translator:**

Reviewers：

①**Background**

In the actual market, customers need some filaments that have higher elasticity than PLA. TPU is a soft material and can be used for printing elastic objects.

②**Main Ingredients**

TPU, color additive.

③**Features**

Flexible material with the same elasticity as rubber.

The thinner and less filled the printed model is, the higher elasticity and softness it gets.

Low viscosity, easy to cooling.

Uvioresistant.

④**Application and Target Audience**

Generally used for printing elastic products, suitable for all groups who want to print soft models.

⑤**Technical Specification**

Filament Diameter: 1.75mm

Tolerance: ±0.05mm

Printing Temperature: 200°C-220°C

Heated Bed Temperature: 0°C-60°C/ without heating

Printing Speed: 15-40mm/s

⑥**Shortcomings**

- TPU is soft material, so the filament can't be rolled neatly.
- The material itself is soft and does not bear any force. It is recommended to use the direct drive extruder to get a good printing experience. The remote extruder can only print normally after modification.
- Bowden extruder is difficult to use and requires high level of users.

⑦**Relevant Parameters of Recommended Machine Types**

APPENDIX 3

Eryone TPU filament data sheet (Part 2)

Relevant Parameters of Recommended Machine Types		
Type	Extruder Type/Heated Bed Type	Parameter
Prusa i3	Direct Drive Extruder/PEI Bed Sticker	Printing Temperature: 200-210℃ Heated Bed Temperature: 0-50℃/without heating Printing Speed: 15-30mm/s Retracting Length: 0.8mm Retracting Speed: 30-40mm/s

Basic Parameters

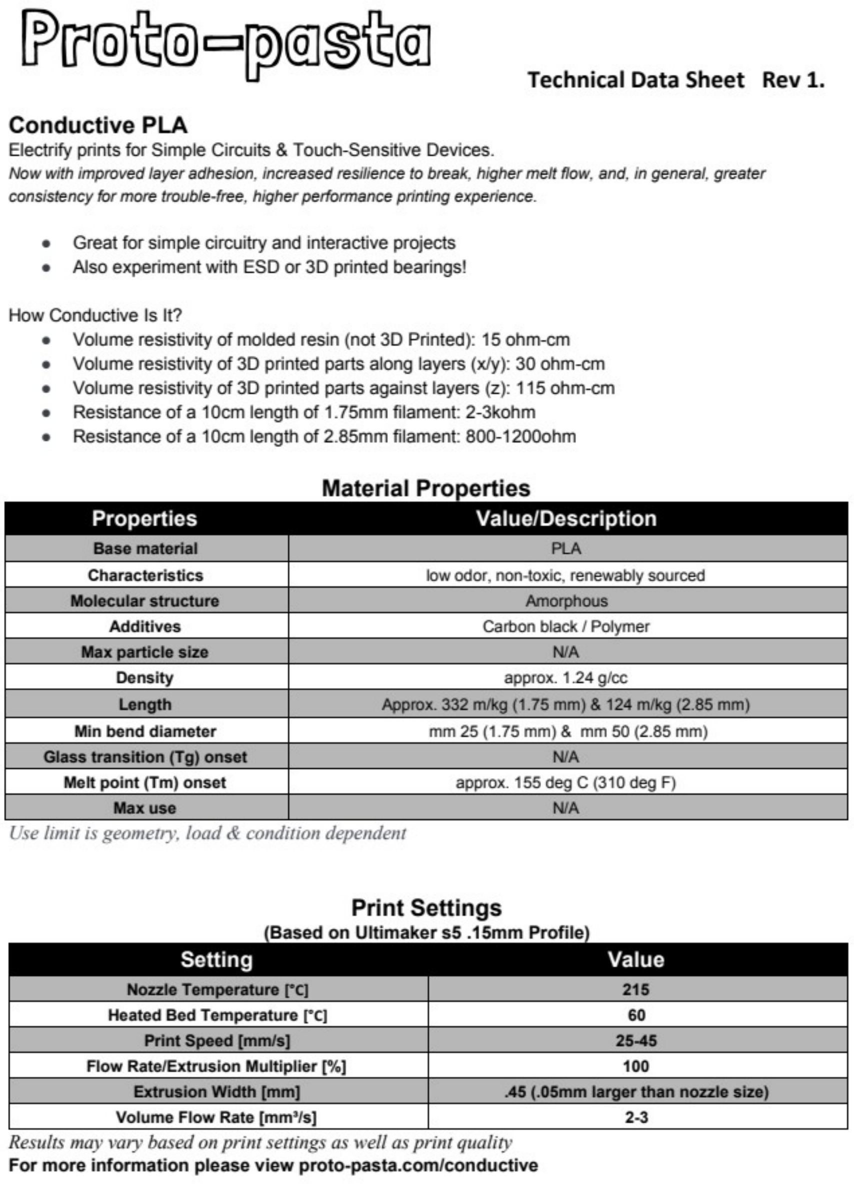
TPU Basic Parameter		
Physical Properties	Typical Value	Method
Specific Gravity [g/cm^3]	1.21	ISO 1183
Moisture Absorption 24 h [%] (2)	/	/
Moisture Absorption 7 day [%] (2)	/	/
Moisture Absorption 4 weeks [%] (2)	/	/
Heat Deflection Temperature (0,45MPa)	/	ISO 75
Tensile Yield Strength Filament [MPa]	40	ASTM D412
Shore Hardness	95A	
Explain		
(1) 2.16kg; 210℃		
(2) 28℃; humidity: 37%		

Table 1

Eryone. Eryone TPU technical data sheet. (2023). Retrieved March 11, 2023, from: https://cdn.shopifycdn.net/s/files/1/0252/0412/9841/files/Specification_of_TPU_EN.docx?v=1611124393

APPENDIX 4

Protopasta Conductive filament data sheet



Proto-pasta

Technical Data Sheet Rev 1.

Conductive PLA

Electrify prints for Simple Circuits & Touch-Sensitive Devices.

Now with improved layer adhesion, increased resilience to break, higher melt flow, and, in general, greater consistency for more trouble-free, higher performance printing experience.

- Great for simple circuitry and interactive projects
- Also experiment with ESD or 3D printed bearings!

How Conductive Is It?

- Volume resistivity of molded resin (not 3D Printed): 15 ohm-cm
- Volume resistivity of 3D printed parts along layers (x/y): 30 ohm-cm
- Volume resistivity of 3D printed parts against layers (z): 115 ohm-cm
- Resistance of a 10cm length of 1.75mm filament: 2-3kohm
- Resistance of a 10cm length of 2.85mm filament: 800-1200ohm

Material Properties

Properties	Value/Description
Base material	PLA
Characteristics	low odor, non-toxic, renewably sourced
Molecular structure	Amorphous
Additives	Carbon black / Polymer
Max particle size	N/A
Density	approx. 1.24 g/cc
Length	Approx. 332 m/kg (1.75 mm) & 124 m/kg (2.85 mm)
Min bend diameter	mm 25 (1.75 mm) & mm 50 (2.85 mm)
Glass transition (Tg) onset	N/A
Melt point (Tm) onset	approx. 155 deg C (310 deg F)
Max use	N/A

Use limit is geometry, load & condition dependent

Print Settings
(Based on Ultimaker s5 .15mm Profile)

Setting	Value
Nozzle Temperature [°C]	215
Heated Bed Temperature [°C]	60
Print Speed [mm/s]	25-45
Flow Rate/Extrusion Multiplier [%]	100
Extrusion Width [mm]	.45 (.05mm larger than nozzle size)
Volume Flow Rate [mm³/s]	2-3

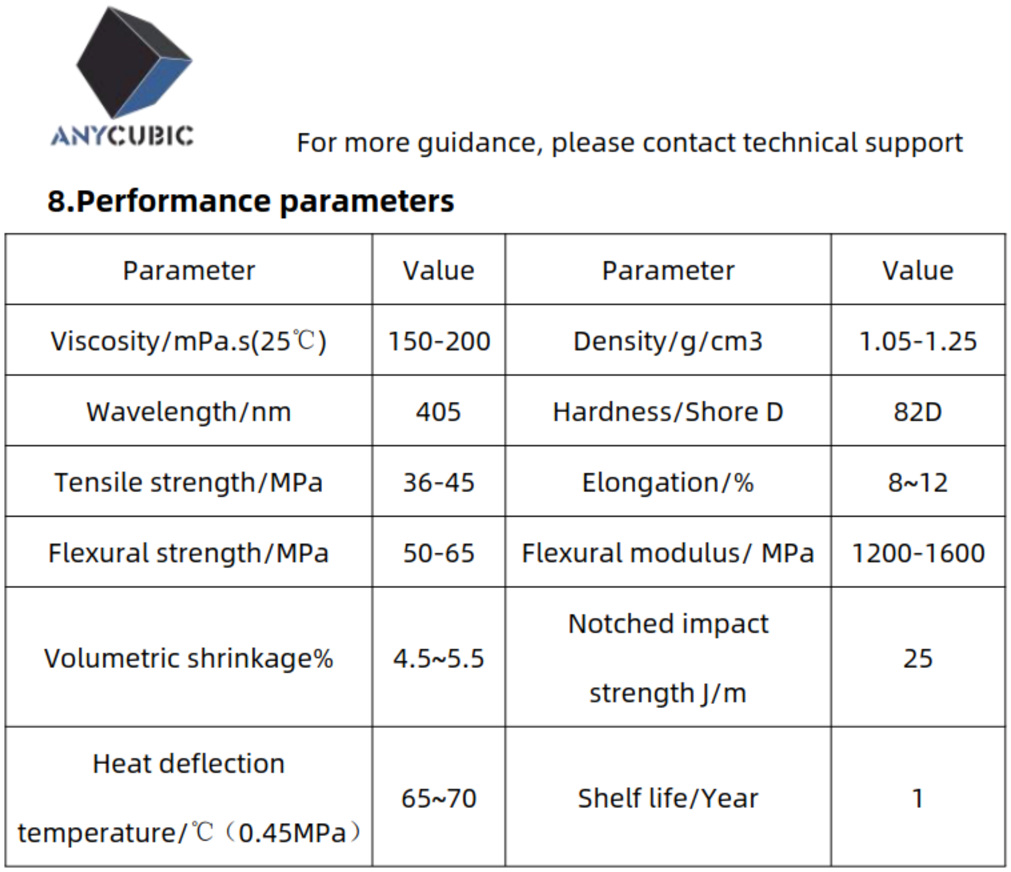
Results may vary based on print settings as well as print quality

For more information please view proto-pasta.com/conductive

Protopasta. Protopaste Conductive technical data sheet. (2023). Retrieved March 11, 2023, from: https://cdn.shopify.com/s/files/1/0717/9095/files/TDS__Conductive_PLA_1.0.1.pdf?1771

APPENDIX 5

Anycubic resin data sheet



ANYCUBIC

For more guidance, please contact technical support

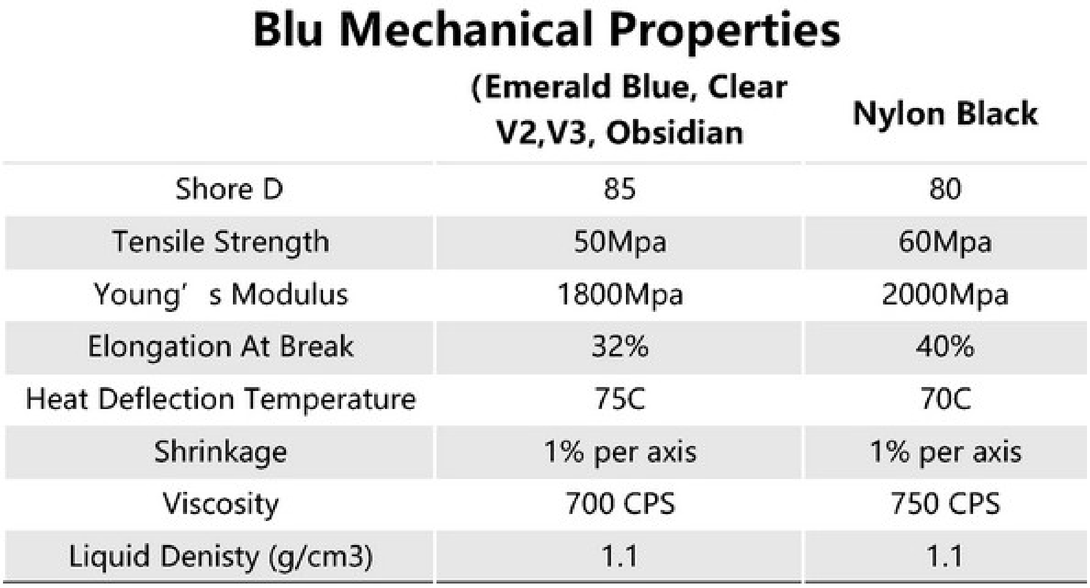
8.Performance parameters

Parameter	Value	Parameter	Value
Viscosity/mPa.s(25℃)	150-200	Density/g/cm3	1.05-1.25
Wavelength/nm	405	Hardness/Shore D	82D
Tensile strength/MPa	36-45	Elongation/%	8~12
Flexural strength/MPa	50-65	Flexural modulus/ MPa	1200-1600
Volumetric shrinkage%	4.5~5.5	Notched impact strength J/m	25
Heat deflection temperature/℃（0.45MPa）	65~70	Shelf life/Year	1

Anycubic. Anycubic resin technical data sheet. (2023). Retrieved March 11, 2023, from: **https://cdn.shopify.com/s/files/1/0245/5519/2380/files/Anycubic_Standard_Resin_User_Manual_V1.0-EN_1.pdf?v=1663574587**

APPENDIX 6

Siraya Tech Blu resin data sheet

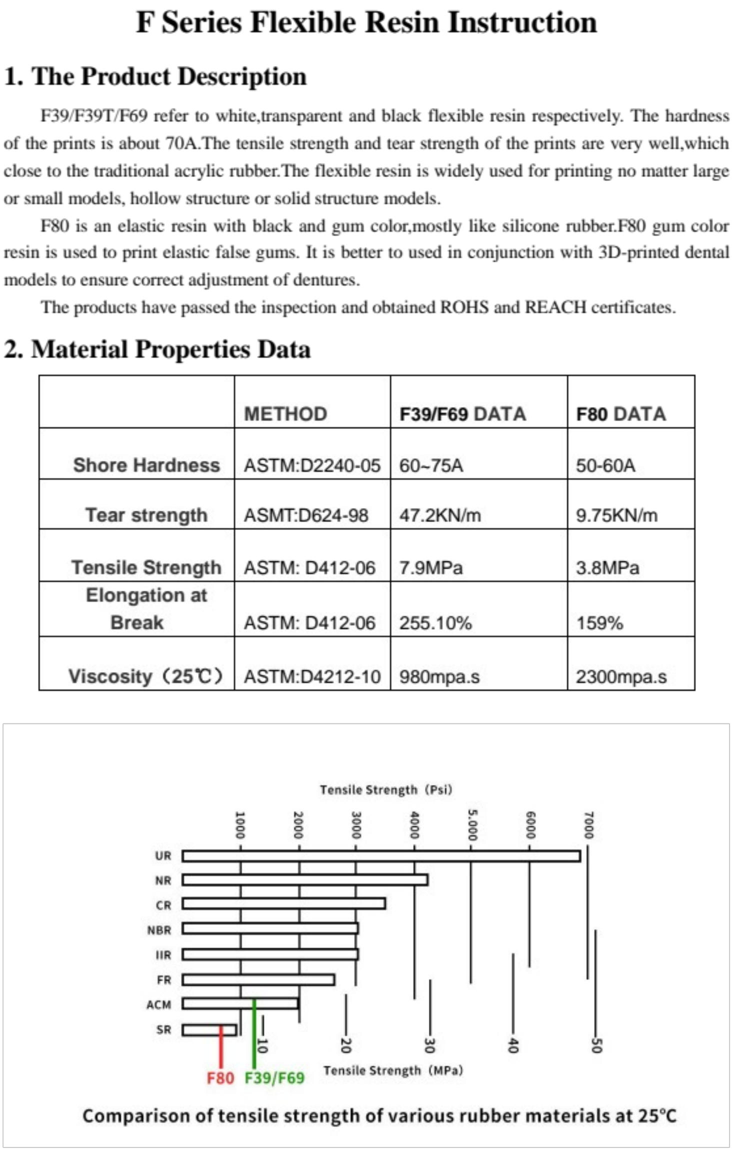


Blu Mechanical Properties

	(Emerald Blue, Clear V2,V3, Obsidian	Nylon Black
Shore D	85	80
Tensile Strength	50Mpa	60Mpa
Young’ s Modulus	1800Mpa	2000Mpa
Elongation At Break	32%	40%
Heat Deflection Temperature	75C	70C
Shrinkage	1% per axis	1% per axis
Viscosity	700 CPS	750 CPS
Liquid Denisty (g/cm3)	1.1	1.1

Siraya. Siraya Tech Blu resin technical data sheet. (2023). Retrieved March 11, 2023, from: https://siraya.tech/products/blu-tough-resin-by-siraya

Resione F80 Resin data sheet



F Series Flexible Resin Instruction

1. The Product Description

F39/F39T/F69 refer to white,transparent and black flexible resin respectively. The hardness of the prints is about 70A.The tensile strength and tear strength of the prints are very well,which close to the traditional acrylic rubber.The flexible resin is widely used for printing no matter large or small models, hollow structure or solid structure models.

F80 is an elastic resin with black and gum color,mostly like silicone rubber.F80 gum color resin is used to print elastic false gums. It is better to used in conjunction with 3D-printed dental models to ensure correct adjustment of dentures.

The products have passed the inspection and obtained ROHS and REACH certificates.

2. Material Properties Data

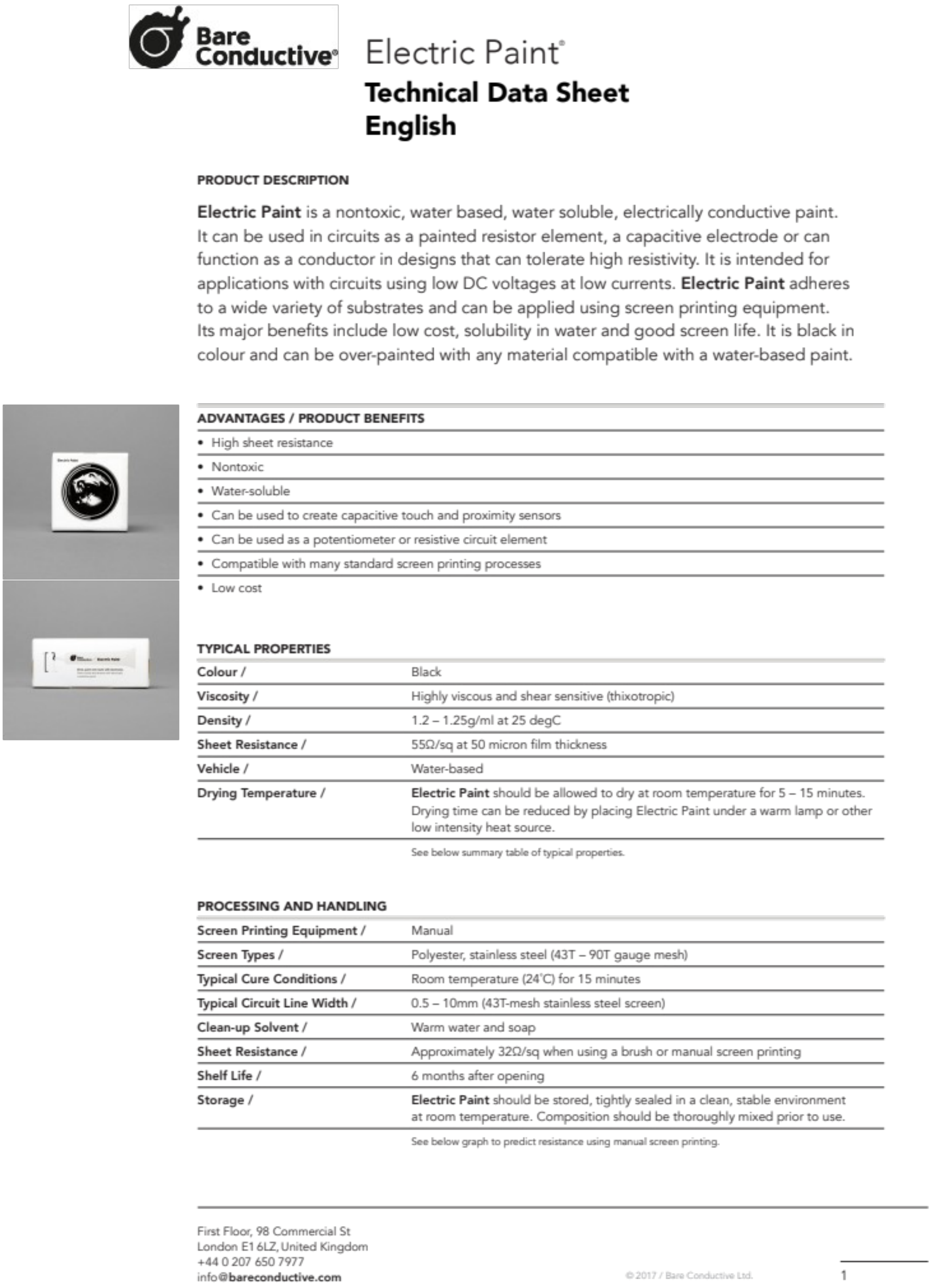
	METHOD	F39/F69 DATA	F80 DATA
Shore Hardness	ASTM:D2240-05	60~75A	50-60A
Tear strength	ASMT:D624-98	47.2KN/m	9.75KN/m
Tensile Strength	ASTM: D412-06	7.9MPa	3.8MPa
Elongation at Break	ASTM: D412-06	255.10%	159%
Viscosity（25℃）	ASTM:D4212-10	980mpa.s	2300mpa.s

Comparison of tensile strength of various rubber materials at 25℃

Resione. Resione F80 Resin technical data sheet. (2023). Retrieved March 11, 2023, from: https://cdn.shopify.com/s/files/1/0464/1033/4360/files/F_series_flexible_elastic_resin_instruction_31aa5e14-4a13-48b4-aaf4-2fd806244a29.pdf?v=1651203409

APPENDIX 8

Bare Conductive Electric Paint data sheet



Bare Conductive

Electric Paint®

Technical Data Sheet
English

PRODUCT DESCRIPTION

Electric Paint is a nontoxic, water based, water soluble, electrically conductive paint. It can be used in circuits as a painted resistor element, a capacitive electrode or can function as a conductor in designs that can tolerate high resistivity. It is intended for applications with circuits using low DC voltages at low currents. **Electric Paint** adheres to a wide variety of substrates and can be applied using screen printing equipment. Its major benefits include low cost, solubility in water and good screen life. It is black in colour and can be over-painted with any material compatible with a water-based paint.

ADVANTAGES / PRODUCT BENEFITS

- High sheet resistance
- Nontoxic
- Water-soluble
- Can be used to create capacitive touch and proximity sensors
- Can be used as a potentiometer or resistive circuit element
- Compatible with many standard screen printing processes
- Low cost

TYPICAL PROPERTIES

Colour /	Black
Viscosity /	Highly viscous and shear sensitive (thixotropic)
Density /	1.2 – 1.25g/ml at 25 degC
Sheet Resistance /	55Ω/sq at 50 micron film thickness
Vehicle /	Water-based
Drying Temperature /	**Electric Paint** should be allowed to dry at room temperature for 5 – 15 minutes. Drying time can be reduced by placing Electric Paint under a warm lamp or other low intensity heat source.

See below summary table of typical properties.

PROCESSING AND HANDLING

Screen Printing Equipment /	Manual
Screen Types /	Polyester, stainless steel (43T – 90T gauge mesh)
Typical Cure Conditions /	Room temperature (24˚C) for 15 minutes
Typical Circuit Line Width /	0.5 – 10mm (43T-mesh stainless steel screen)
Clean-up Solvent /	Warm water and soap
Sheet Resistance /	Approximately 32Ω/sq when using a brush or manual screen printing
Shelf Life /	6 months after opening
Storage /	**Electric Paint** should be stored, tightly sealed in a clean, stable environment at room temperature. Composition should be thoroughly mixed prior to use.

See below graph to predict resistance using manual screen printing.

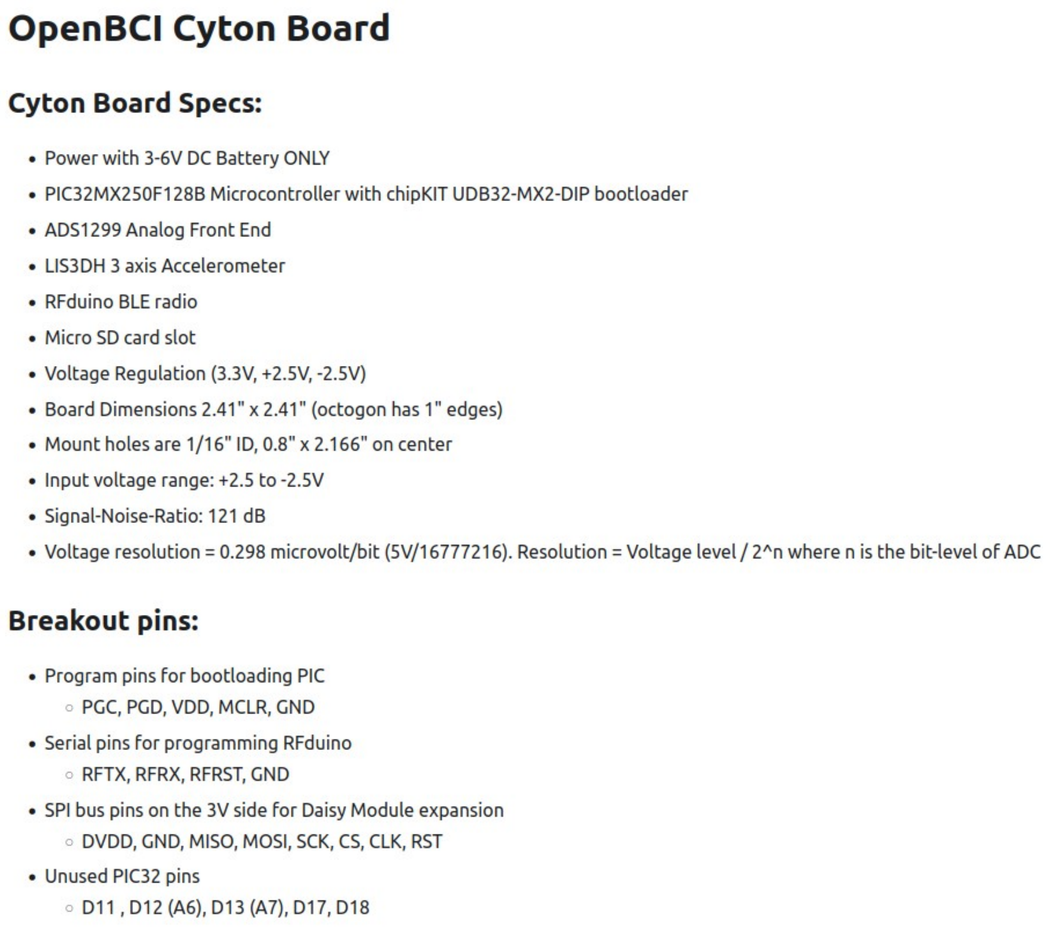
First Floor, 98 Commercial St
London E1 6LZ, United Kingdom
+44 0 207 650 7977
info@**bareconductive.com**

© 2017 / Bare Conductive Ltd.

1

Bare Conductive. Bare Conductive Paint technical data sheet. (2023). Retrieved March 11, 2023, from: https://cdn.shopify.com/s/files/1/0520/3669/8292/files/EP_tech_data_sheet_a13f3d46-56ce-4689-97cb-da3b3b3d52d2.pdf?v=1655713221

OpenBCI Cyton board data sheet



OpenBCI Cyton Board

Cyton Board Specs:

- Power with 3-6V DC Battery ONLY
- PIC32MX250F128B Microcontroller with chipKIT UDB32-MX2-DIP bootloader
- ADS1299 Analog Front End
- LIS3DH 3 axis Accelerometer
- RFduino BLE radio
- Micro SD card slot
- Voltage Regulation (3.3V, +2.5V, -2.5V)
- Board Dimensions 2.41" x 2.41" (octogon has 1" edges)
- Mount holes are 1/16" ID, 0.8" x 2.166" on center
- Input voltage range: +2.5 to -2.5V
- Signal-Noise-Ratio: 121 dB
- Voltage resolution = 0.298 microvolt/bit (5V/16777216). Resolution = Voltage level / 2^n where n is the bit-level of ADC

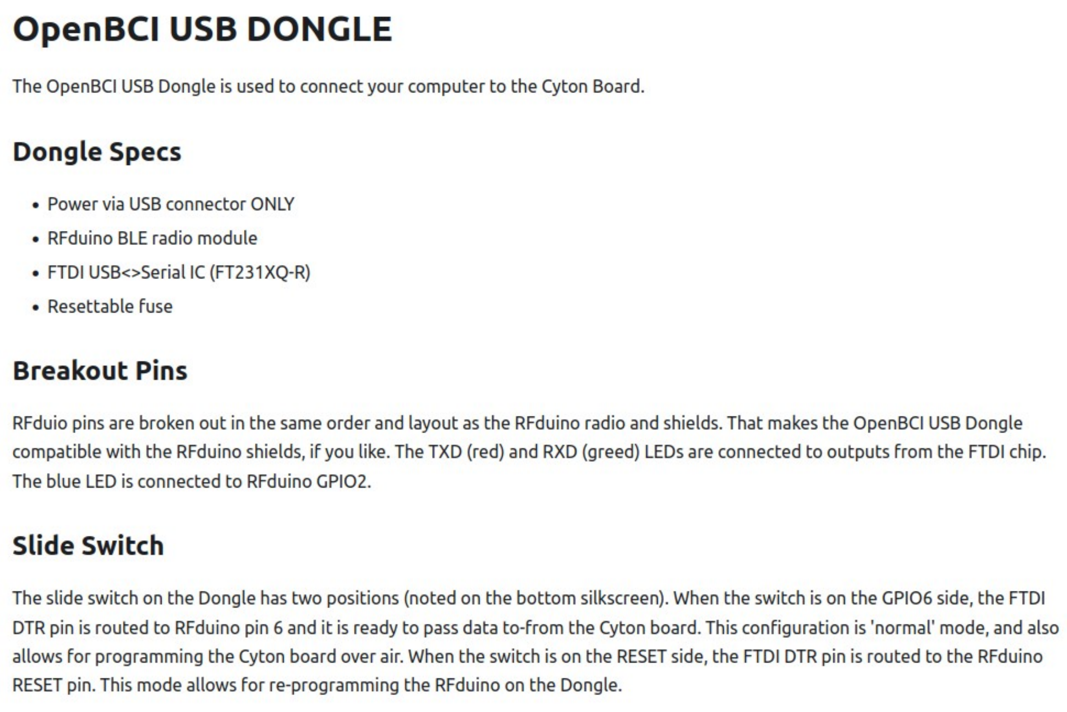
Breakout pins:

- Program pins for bootloading PIC
  - PGC, PGD, VDD, MCLR, GND
- Serial pins for programming RFduino
  - RFTX, RFRX, RFRST, GND
- SPI bus pins on the 3V side for Daisy Module expansion
  - DVDD, GND, MISO, MOSI, SCK, CS, CLK, RST
- Unused PIC32 pins
  - D11 , D12 (A6), D13 (A7), D17, D18

OpenBCI. OpenBCI Cyton board data sheet. (2023). Retrieved March 11, 2023, from: https://docs.openbci.com/Cyton/CytonSpecs/#cyton-board-specs

APPENDIX 10

OpenBCI USB Dongle data sheet



OpenBCI USB DONGLE

The OpenBCI USB Dongle is used to connect your computer to the Cyton Board.

Dongle Specs

- Power via USB connector ONLY
- RFduino BLE radio module
- FTDI USB<>Serial IC (FT231XQ-R)
- Resettable fuse

Breakout Pins

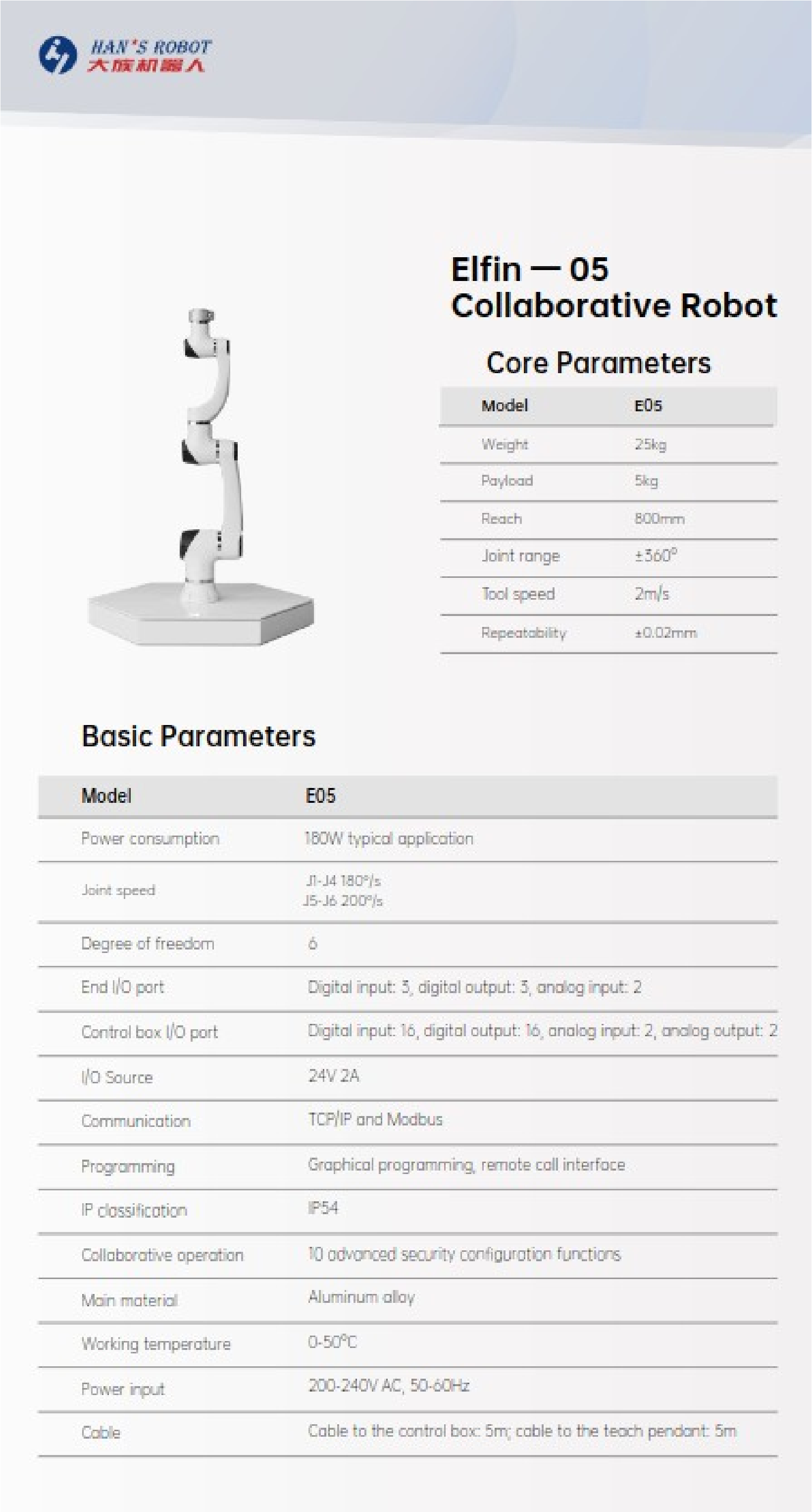
RFduio pins are broken out in the same order and layout as the RFduino radio and shields. That makes the OpenBCI USB Dongle compatible with the RFduino shields, if you like. The TXD (red) and RXD (greed) LEDs are connected to outputs from the FTDI chip. The blue LED is connected to RFduino GPIO2.

Slide Switch

The slide switch on the Dongle has two positions (noted on the bottom silkscreen). When the switch is on the GPIO6 side, the FTDI DTR pin is routed to RFduino pin 6 and it is ready to pass data to-from the Cyton board. This configuration is 'normal' mode, and also allows for programming the Cyton board over air. When the switch is on the RESET side, the FTDI DTR pin is routed to the RFduino RESET pin. This mode allows for re-programming the RFduino on the Dongle.

OpenBCI. OpenBCI USB Dongle data sheet. (2023). Retrieved March 11, 2023, from: https://docs.openbci.com/Cyton/CytonSpecs/#cyton-board-specs

Elfin5 Robot Arm data sheet



HAN'S ROBOT 大族机器人

Elfin — 05 Collaborative Robot

Core Parameters

Model	E05
Weight	25kg
Payload	5kg
Reach	800mm
Joint range	±360°
Tool speed	2m/s
Repeatability	±0.02mm

Basic Parameters

Model	E05
Power consumption	180W typical application
Joint speed	J1-J4 180°/s J5-J6 200°/s
Degree of freedom	6
End I/O port	Digital input: 3, digital output: 3, analog input: 2
Control box I/O port	Digital input: 16, digital output: 16, analog input: 2, analog output: 2
I/O Source	24V 2A
Communication	TCP/IP and Modbus
Programming	Graphical programming, remote call interface
IP classification	IP54
Collaborative operation	10 advanced security configuration functions
Main material	Aluminum alloy
Working temperature	0-50°C
Power input	200-240V AC, 50-60Hz
Cable	Cable to the control box: 5m; cable to the teach pendant: 5m

Hans Robot. Elfin5 Robot Arm data sheet. (2023). Retrieved March 11, 2023, from: https://www.hansrobot.net/service/download/user-manual