EEG/EMG-BASED HARDWARE MOBILIZING

Senior Design Project II

Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz

2022

MEF UNIVERSITY
FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

EEG/EMG-BASED HARDWARE MOBILIZING

Senior Design Project II

Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz

Advisors: Dr. Tuna Çakar, Assoc. Prof. Şefik Şuayb Arslan

2022

MEF UNIVERSITY
FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

Project Title : EEG/EMG-Based Hardware Mobilizing
Student(s) Name : Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz
Date : 29/05/2022

I hereby state that the design project prepared by **Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz** has been completed under my supervision. I accept this work as a "Senior Design Project".

29/05/2022
Dr. Tuna Çakar

I hereby state that I have examined this senior design project by **Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz**. This work is acceptable as a "Senior Design Project".

29/05/2022
Prof. Dr. Muhittin Gökmen

Head of the Department of
Computer Engineering

ACADEMIC HONESTY PLEDGE

In keeping with MEF University Student Code of Conduct, I pledge that this work is my own and that I have not received inappropriate assistance in its preparation. I further declare that all resources are explicitly cited.

NAME	DATE	SIGNATURE
Burak Bahir Günden	29/05/2022	
Efe Ertekin	29/05/2022	
Yasin Yılmaz	29/05/2022	

ABSTRACT

EEG/EMG-BASED HARDWARE MOBILIZING

Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz

MEF UNIVERSITY
Faculty of Engineering
Department of Computer Engineering

Advisors: **Dr. Tuna Çakar, Assoc. Prof. Şefik Şuayb Arslan**

MAY, 2022

In this project, the main scope is to develop a brain-computer interface (BCI) with the use of PiCar and EEG/ERP devices. The ultimate goal of this project is to direct and control a PiCar with respect to the signals captured via the EEG/ERP device. With the EEG headset, we obtain the signals of mental/motor imagery and observe gestures as EMG of a person. With the collected data, filtering and other methods are applied to have noise-free signals. With that data, machine learning models are trained. Models are used to predict the direction that is passed as an input to PiCar's API to move the PiCar according to the result.

Keywords: EEG, Raspberry Pi, Brain Wave Signals, OpenBCI, Machine Learning, EMG, ERP, PiCar

ÖZET

EEG/EMG TABANLI DONANIM MOBİLİZASYONU

Burak Bahir Günden, Efe Ertekin, Yasin Yılmaz

MEF ÜNİVERSİTESİ
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Tez Danışmanları: **Dr. Tuna Çakar, Assoc. Prof. Şefik Şuayb Arslan**

MAYIS, 2022

Bu projede ana kapsam, PiCar ve EEG/ERP cihazlarının kullanımı ile bir beyin-bilgisayar arayüzü (BCI) geliştirmektir. Bu projenin nihai amacı, EEG/ERP cihazı aracılığıyla yakalanan sinyallere göre bir PiCar'ı yönlendirmek ve kontrol etmektir. EEG seti ile bir kişinin zihinsel/motor imgeleme sinyallerini ve mimik yaparken yarattığı sinyalleri toplarız. Toplanan verilerle, gürültüsüz sinyallere sahip olmak için filtreleme ve diğer yöntemler uygulanır. Bu verilerle makine öğrenimi modelleri eğitilir. Modeller, sonuca göre PiCar'ı hareket ettirmek için PiCar'ın API'sine girdi olarak geçirilen yönü tahmin etmek için kullanılır.

Anahtar Kelimeler: EEG, Raspberry Pi, Beyin Sinyalleri, OpenBCI, Makine Öğrenmesi, EMG, ERP, PiCar

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
BCI	Brain-Computer Interface
EEG	Electroencephalogram
ERP	Event-Related Potential
GUI	Graphical User Interface
CNN	Convolutional Neural Network
SDP	Senior Design Project
EMG	Electromyography
SVM	Support Vector Machines
LR	Logistic Regression
LDA	Linear Discriminant Analysis
RFT	Random Forest Tree
GBC	Gradient Boosting Classification
KNN	K-Nearest Neighbors Classification
DTC	Decision Tree Classification
VC	Voting Classification
MNB	Multinomial Naive Bayesian

1. INTRODUCTION

In recent years, artificial intelligence has been developed rapidly within its increasing usage and now it is more capable of creating solutions to real-life problems and it has a great impact on everyone's life. The creation of the Brain-Computer Interfaces (BCI) made it possible for people to work on ERP and EEG data. This technology enables researchers to work on data that is a creation from the electrical activity of the human brain. The motivation behind the project is to come up with an artificial intelligence solution for people who have disabilities and diseases. The problem that is expected to solve in the scope of the project is detecting the imagined direction and performing a gesture that is already assigned for a direction with the help of a BCI. The solution to this problem includes the following.

The data will be obtained using a headset (BCI) that includes 16 electrodes. The numeric values of electrical activity obtained by electrodes will be transmitted to a computer from a BCI device and a library called Brainflow will be used to handle the real-time data in a python program. The data will be cleansed using signal processing methods. A user interface will be designed for obtaining the data required to train the model. This motor imagery UI will consist of gifs related to the direction and its integration with the Brainflow scripts will be ensured. On the other hand, gestures UI will also have gifs related to the direction and the same operations will be performed on this as well.

After collecting the required data, the models will be trained with the collected datasets. Multinomial classification algorithms will be used like KNN, Naive Bayes, Decision Tree, SVM, and so on. To increase the performance of the models, different methods will be used like outlier removal, voting classifier, etc. And for both systems, different parameters will be given for algorithms to achieve the best score. For this, GridSearch will be used and with its help, models will have better results in every aspect. Models will be tested with subjects and observations will be made.

1.1. Motivation

The main goal in the development of this project is to make it easier for amputees and people with different kinds of physical problems to use their prostheses. Performing these operations over the brain, which is an organ that can be used by anyone conscious, adds universality to this work. With this project, it will be possible to minimize the time of developing personalized prostheses and realize faster prosthetic adaptations.

1.2. Broad Impact

The broader impact of the project is to facilitate the daily activities of people with disabilities.

1.2.1. Global Impact of the solution

First of all, it is very possible that this project will have large-scale effects. By great impact, we mean it can have a positive impact on the lives of many people. But we will talk about these in more detail in the social effects section. This project, which is likely to have a global impact, can be a salable product when necessary developments and support are provided, and it may be possible to sell it cheaply to people in need all over the world.

1.2.2. Economic Impact of the solution

Economically, it is not possible to measure the extent of the effects of this project. If an indirect effect is considered, the effects of the people that this project will bring to life can be mentioned. This project is suitable for evolving into another field. According to this flexibility, employment opportunities may occur.

1.2.3. Environmental Impact of the solution

It is possible to say that their contribution to the environment is positive, as a lot of work can be done with less equipment.

1.2.4. Societal Impacts of the solution

The contribution of this project to society is too great to be denied. It is aimed that people with physical problems or suffering from different diseases will adapt to life and turn them into productive individuals with such projects. For example, it may be possible for someone in a wheelchair to steer their own chair with this device after the necessary motor equipment is added to their vehicle.

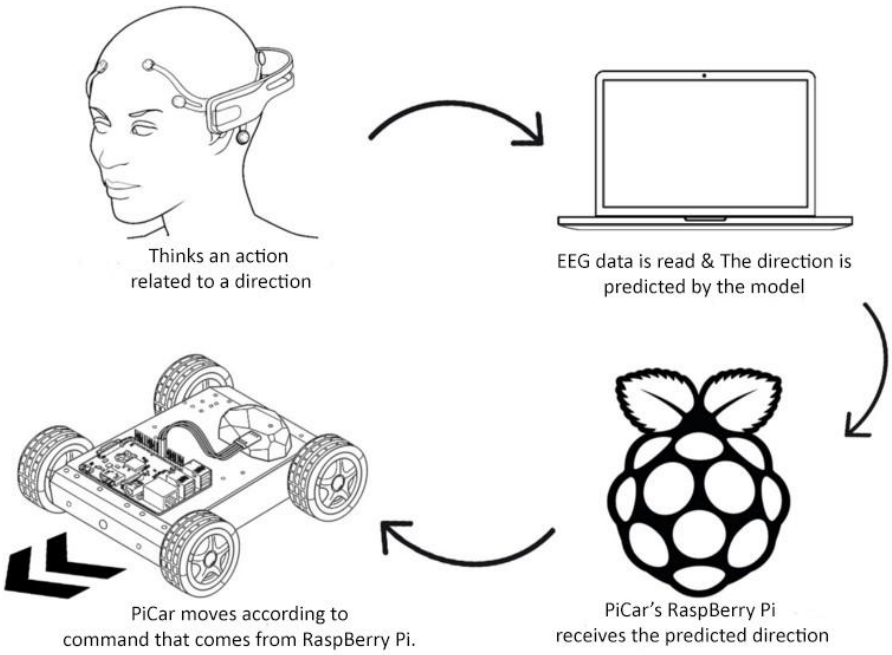
1.2.5. Legal Issues related to the project

To talk about the legal effects, there may be cases of malicious use of these systems. In such cases, the operation of the system should be prevented. Otherwise, it will be possible for people who have integrated these systems into their lives to experience some problems. If we foresee that people who are already dealing with certain problems due to the nature of this system will be users, it will be possible to say that it will not be a pleasant situation for both them and the developers of this system.

2. PROJECT DEFINITION AND PLANNING

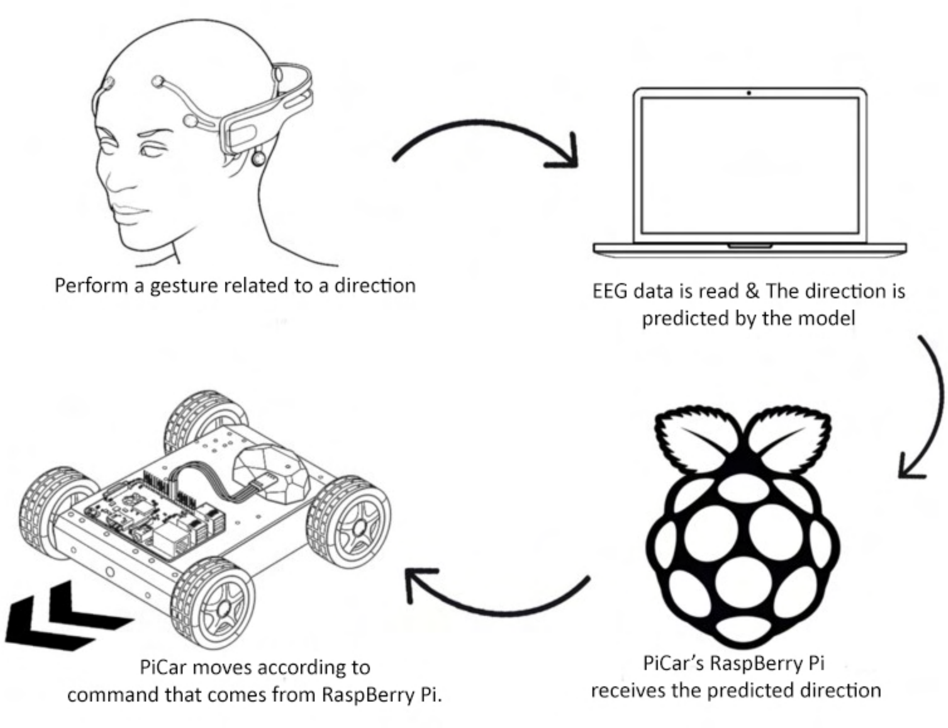
2.1. Project Definition

In order to realize the project, BCI (Brain-Computer Interface) product must be used. This product, which is required for the use of EEG signals, plays a role in the visualization of the electrical output of the neurons of the brain. In order to collect and transform these data into meaningful data, classification is made using AI and ML algorithms. Models are trained and tests are conducted. For both systems mentioned, we have dreamed of an F1 score and accuracy above 70%. Of course, valid for every different direction. While we focus on the first system that makes this possible and set that system to work correctly in this 1.5-month period, we also wanted to ensure that the other system works in an average way or is prepared in a way that can be run in the future. We aim to train personalized systems. Thus, it is going to be possible to achieve higher accuracy and we wouldn't have to deal with the problems of eliminating the differences created by thinking between different people. It is a difficult task to follow with the equipment we have that brain signals differ in people even in the same events. After the creation of different models, they are tested in both mathematical and real-life experiments.



Figure#1: General working principle of the Mental Imagery (EEG) system

The general flow is shown in Figure#1. To explain this in detail, first of all, it is necessary to talk about the existence of a data collection process. In this section, a python code has been written to collect data and data is collected with gifs that appear on the screen instantly. There are 5 of these gifs in total. Forward, backward, right, left, and stop. All of them are given 5 seconds to train and 0.5 seconds are expected during the transition (times can vary). In order for the data collected here to be more accurate, cutting a period of 1 second from the beginning will be tested during the development of the models and an action will be taken accordingly. It may be possible to see a difference in the durations.



Figure#2: General working principle of gesture (EMG) system

It is possible to observe a similar system in Figure#2. In this system, action observation is made and it is possible to obtain higher accuracy and F1 scores. Facial mimics or gestures provide a great convenience in providing these. It has been proven in some studies in the theoretical background that this system can work.

Models trained with different algorithms will be tested and the best ones will continue. It is not very realistic to determine these algorithms in advance, but it is possible to say that algorithms such as KNN, Naive Bayes, Decision Tree, SVM, Logistic Regression, Random Forest Classifier, etc. will be tested. We already know that it is possible for different algorithms to give better results for both systems, and we also think that algorithms with different parameters should be used to give better results. For this, GridSearch will be used and appropriate parameters will be found for each algorithm. Thus, different parameters will be obtained for two different systems.

The usage areas of the project are very wide. It is primarily developed for disabled individuals and those with physical illnesses, and it will be designed to be used by people who do not have any additional medical problems. In other words, it will be possible to establish a system almost as if an extra limb has been added to people. We would like to state that we did not specifically work on this, but we did not want to go without saying that this is possible.

One of the biggest risks of the project is the failures that will occur in the equipment. Repairs like these are both time-consuming and require a lot of hardware knowledge. In the event that the EEG signals cannot be collected properly due to the breakdown of the device, it is aimed to continue the project by trying different ways. If that part of the device cannot be repaired, it can be continued with the ACC part. If that part is also problematic, a new device will be expected to arrive. Plans are also made in case such problems occur.

2.2. Project Planning

Task	Responsible Person	Weeks													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Review Previous Work	Burak Efe Yasin														
Discuss the design of the architecture	Burak Efe Yasin														
Coding interface to collect the data	Burak Efe Yasin														
Collect the data and train the model	Burak Efe Yasin														
Improve current models and add new models	Burak Efe Yasin														
Testing & Troubleshooting	Burak Efe Yasin														
Report and Project Presentation	Burak Efe Yasin														

Table 1. Project Plan for Semester

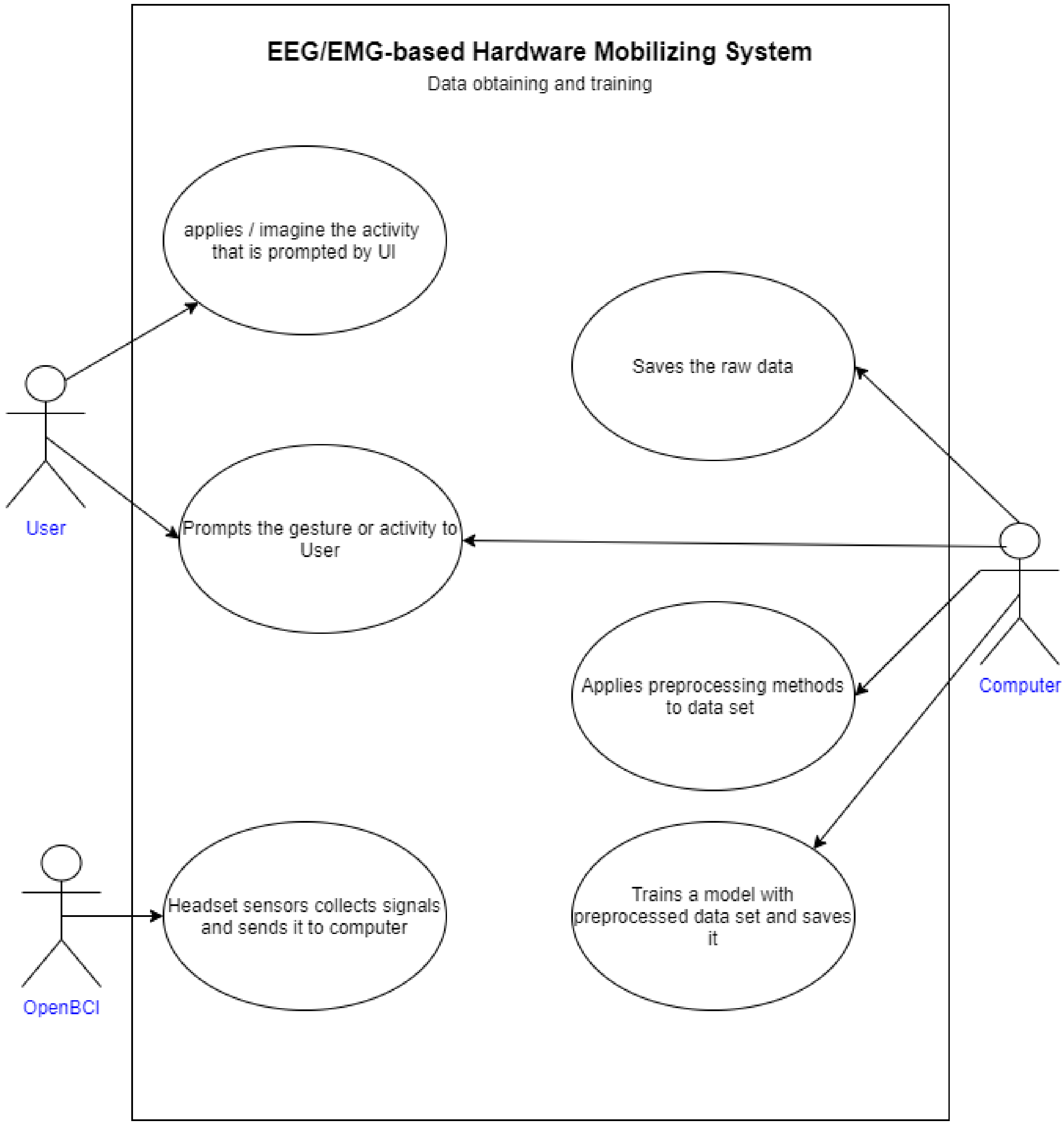
2.2.1 Aim of the Project

The project aims to provide software that enables the communication between two distinct devices. This software acts as a bridge between OpenBCI and Raspberry Pi. This software enables the implementation of brain signal detection to give mobilizing to hardwares in this case PiCar, which is a drone-car. Two different methods are used to obtain data from OpenBCI's EEG headset. EEG and EMG based two different ways are performed. With that data, models are trained and used to predict the direction. After that, directions are used to move the drone-car.

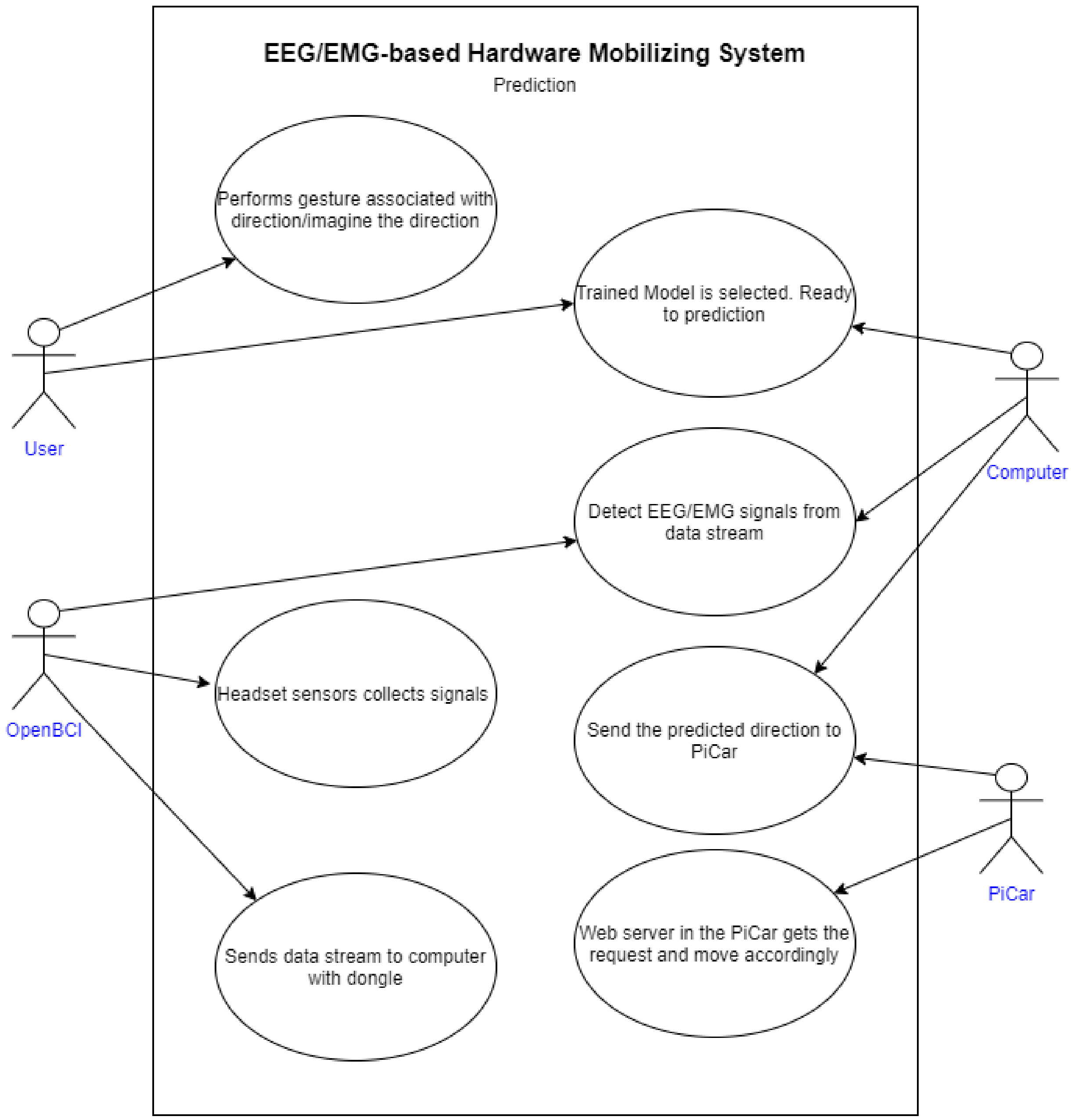
2.2.2 Project Coverage

The project consists of multiple stages. First of all, data must be collected. But how data is collected is important. It needs to be filtered to get rid of the noise that is created by the environment, and after that, it should be divided into many pieces to create more features for machine learning algorithms. More pieces are specifically alphas, betas, gammas, and thetas. They are representing different Hz gaps of the collected data. With the live-streaming sessions, these features are created and models predict the direction with the provided data that comes from the streaming session.

2.2.3 Use Cases



Figure#3: Use Case Diagram of Data Obtaining and Training



Figure#4: Use Case Diagram of Prediction

2.2.4 Success Criteria

The succes is relied on the efficiency of machine learning models and success on integration between all devices in the system and the pipeline which includes data obtaining, preprocessing, classifying and predicting flow.

2.2.5 Project Time and Resource Estimation

We met 2 times a week in the neuroscience laboratory available at the university. All the hardware tools used are provided by this laboratory. In each meeting, we stay at least 4 hours in order to make progress. There were also online meetings and they were about 3-9 hours long.

The prices of the hardware tools are as follows.
All-in-One Biosensing R&D Bundle: $3199.99
SunFounder PiCar-V Kit V2.0 for Raspberry Pi with Raspberry Pi: $179.99

2.2.6 Solution Strategies and Applicable Methods

To train the machine learning model, a dataset is mandatory. To create a dataset which relies on the purpose of the project, a UI is created with python. Since we have the BCI headset, it is possible to obtain data.

To filter the signal, the signal processing methods were needed. As a solution to this, filters available in the Brainflow were used. These methods help us to reduce the noise in the signal, as a downside, these methods might result in the signal losing the valuable data it carries.

To observe the data, the data is visualized using the libraries created for Python. There was an imbalance in the supports in the models, it is solved using oversampling. Lastly, For increasing the number of features, channel values in the different band powers are obtained.

Applicable Methods

Prediction: While predicting the direction of the relevant EEG signals simultaneously, the data are obtained every single second. Prediction is made by using a windowing method. The data collected over a period of time is shifted based on the 200ms time limit. In each prediction, the data is shifted and a prediction is made. After the prediction, the probability of each direction retrieved by the model is gathered and the prediction with the greatest probability is chosen. If the greatest prediction made exceeds the threshold. The predicted label is accepted as the direction that is imagined.

Feature Engineering: The common spatial patterns (CSP) algorithm is widely used in the feature engineering of EEG datasets. It can be implemented

Enlarging the dataset: The dataset can be largened by using more people in the data obtaining sessions. As a result of this accuracy and the other metrics might result better. Also, before applying these processes, it is possible to create more values such as alpha and beta with the data in hand, with a special method. This method works like this: For example, you have 500 units of data. In order to create an alpha value from this data, you need to round it as 100-100-100-100-100. Thus, you will be able to get 5 different alpha values. However, instead of doing this, more data is created by progressing step by step, that is, between 0-100, between 25-125, and between 50-150. The jump number here, 25, may vary. This method may cause the dataset to overfit.

2.2.7 Risk Analysis

If the hardware tools used in the project get damaged or become unusable, the tools have to be supplied again. The devices used are not easy to access and also not cheap. So in a scenario that it becomes unusable, we will have to wait until we get the new devices and it will cost time for us because we will not be able to make any progress until the devices arrive.

Also, The dataset is obtained by the Cython+Daisy headset that we have. In the worst-case scenario, we would not be able to create a dataset.

2.2.8 Tools Needed

The software and hardware tools used in the project are as follows:

Hardware Tools:

Ultracortex Mark IV: It is a 3D printed headset for working with OpenBCI. This kit consists of Frames, cables, spikey units, flat units, comfort units, ear clips, and an OpenBCI board.

Frame: The frame is the body of the headset which is 3-d printable.

Spikey units: These electrodes are UltraCortex nodes (used in the head surface with hair) in the headset UltraCortex. Its quantity changes depending on the number of channels on the board(8 or 16).

Flat units: These electrodes are UltraCortex nodes that are not spiky (not used in the head surface with hair for example forehead).

Comfort units: Its purpose is to distribute the weight of the headset on the head.

Ear clip: These are ear clips electrodes.

Cables: Cables are used to connect the electrodes to the board.

OpenBCI Board: It is a bio-sensing system used to measure and keep track of (or record) electrical activity of the brain(EEG), muscles(EMG), and heart(ECG). It can be used with standard EEG electrodes. In our project, it is going to be used for recording brain(EEG) signals. The specs of the board are given below.

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" (octagon has 1" edges)

- Mount holes are 1/16" ID, 0.8" x 2.166" on center

Dongle: It is a device for establishing a connection between the cython and the computer. It has an integrated RFDuino which communicates with RFDuino on the board. A serial connection is established between the dongle and the computer's onboard FTDI chip. This port is called COM in windows /dev/tty for macOS and Linux

PiCar: It is a robotic car that can work with Raspberry Pi 4 Model B, 3 Model B+/B, 2 Model B, Model B+. After the head motions are detected, These motions will be used for mobilizing the car.

Software Tools:

Brainflow: It is a library used for obtaining and parsing biosensor data such as EEG, EMG and ECG. Brainflow provides an API for BCI headsets in various programming languages. In the project Brainflow's API for python was used.

Django: It is a python framework used for developing web applications. In the project it gets the HTTP request that contains motion from the computer.

PiCar: The PiCar's provider company SunFounder also provides software to move the car. It allows us to use the back wheels to move the car backward and forward and the front wheels to direct the car.

Scikit-learn: To train a model using the dataset, the implementations provided by the scikit-learn library were used. It is also used while applying preprocessing on the dataset.

Autoviz/Seaborn/Matplotlib: These libraries are used for visualizing the dataset.

Pickle: This library is used for saving the trained machine learning model.

3. THEORETICAL BACKGROUND

First of all, we would like to state that one of our purposes in choosing this topic is to help people with disabilities. Although there is no specific amputee situation that this research wants to solve, we want to enable the integration of the developed methods into devices developed for each amputee type. Only 48% of the population can use the prosthesis to its full capacity [1]. So why is this number that low?

It must be said that every amputee has a unique experience. They all have different patterns and data needs to be collected from that amputee to collect the data needed for prostheses to be usable. With methods such as sEMG, the muscles that the amputee can activate can be followed and the prosthesis can be guided accordingly. The differences in the muscles between amputees also necessitate the collection of specific data for each user. But the use of brain signals, which we will focus on, can create a more universal model. We know that people's brains generate similar signals in certain situations. Using this, we want to simplify the classification part (for prosthesis) where every amputee lives.

We believe that people suffering from diseases such as ALS can also make their lives easier. Stephen Hawking, for example, was one of those patients. The majority of these patients are unable to activate their motor neurons [2]. We think that the most appropriate method for developing devices that can help these people is to receive signals from the brain. To give an example, if a system that can move a mouse on a computer is created, it will be possible to make the lives of these people easier.

3.1. Literature Survey

Control of transhumeral prostheses based on electromyography pattern recognition: from amputees to deep learning

Firstly, we need to know what a transhumeral amputation is. Transhumeral amputation is a term that has been used to identify the arm's amputation level. In this amputation type, there is no longer the elbow part of the arm. It is also known as "above elbow amputation". Creating prostheses for every patient is too hard to achieve just because almost all of them are unique.

Currently, myoelectric prostheses are common but they are limited due to their obtaining data methods like EMG (Electromyography).[3] For instance, if the patient loses most of the muscles, these prostheses are almost impossible to implement. Only %48 percent of the people suffering from a transhumeral amputation can use their prostheses with a full capacity. On the first look, this number can be a success but this is not the desired percentage.[3]

How are they performing operations with EMG-based prostheses? Intention creates electrical signals on the muscles. sEMG(surface electromyography) gets that signal and converts it into readable data to make the classification possible. After that, action will be activated for the prosthesis.[3] But sadly, signals differ with each patient. Their muscle signals are unique and it creates a problem that every patient has to create a classification pattern for their prosthesis.[3]

Last 3 decades, machine learning algorithms have been used to handle the classification part of this operation. This way doesn't offer huge improvements anymore. As we now have the ability to access huge datasets and a fast connection to the internet enables us to use DL algorithms to be more accurate in prostheses' movements. Creating the wrong action is way more dangerous than doing nothing at all with prostheses for amputees. CNN or RNN can achieve higher accuracy of classification compared to NN, SVM, etc. So using DL algorithms is more accurate than ML algorithms.[3]

Identification of Raw EEG Signal for Prosthetic Hand Application

EEG(electroencephalogram) is a method that monitors your brain cells' electrical activity. In this project, the goal is to achieve the opening and closing fingers of the prosthetic hand.[4] Equipment is a headset that can capture brain signals with the EEG method, Arduino microcontroller, and lastly, prosthetic hand. They have 2 methods to obtain the EEG. The first one is hand movements, the second one is facial expression technique (eye blink).[4]

They are using an Arduino microcontroller to operate the prosthetic hand. EEG machine (Emotiv epoc+ headset with electrodes) on the other hand creates input for this Arduino microcontroller. They are passing this input via interfaces already developed by them.[4]

In the end, they found the eye blink technique is faster than the hand movement method.[4]

An Approach for Brain-Controlled Prostheses Based on a Facial Expression Paradigm

This project aims to improve brain-controlled prostheses by combining electroencephalography (EEG) signals with facial expressions. The motivation of the project is to improve the efficiency and performance of brain-computer interfaces.[5] There are two equivalent BCI motor-imagery based (MI-BCI) and steady-state visual evoked potential (SSVEP-BCI). In MI-BCI, future research may be restricted by the length of their training and the diversity of their users. The accuracy of MI-BCI is nearly 90% after several months.[5] In SSVEP-BCI, stimulators are used. The most common stimulator is a flashing light pattern. A prolonged stimulation period can easily result in epileptic seizures. Moreover, the accuracy of SSVEP-BCI is 85%.[5]

The used technologies are:

- Portable brain-controlled prosthesis
- Facial-Expression BCI (FE-BCI)
- Arduino Uno L298N
- Intel (R) Core (TM) i5-5600 CPU
- An 8-channel wireless Neuracle manufactured by Neuracle Technology Co., Ltd
- A Bluetooth device to connect Neuracle, microprocessor, and Prosthesis.

The project has nearly 81 ± 5% accuracy for both online and offline experiments on average. Also, this project has no disadvantages like MI-BCI or SSVEP-BCI. The project provides a solid BCI system for patients that have an amputation.[5]

A Self-Learning and Adaptive Control Scheme for Phantom Prosthesis Control Using Combined Neuromuscular and Brain-Wave Bio-Signals

The control scheme of myoelectric prosthesis has a part called pattern recognition. The job of pattern recognition is decoding input data and sending outputs to the myoelectric prosthesis for making the intended gesture. Pattern recognition has a classifier. The classifier is generally calibrated and trained with a supervised learning framework*. [6]

The aim of the project is to develop a better framework for the classifier. In this project, the three-phase identification framework is developed. According to Bio-signals (EMG - EEG), the framework is capable of self-learning. The self-learning framework has higher accuracy on learning patterns and it reduces the lag-time than the supervised learning framework.[6]

*Supervised learning is an ML technique for learning. According to input-output pairs, It maps input to an output.

EEG-Based Hand Motion Pattern Recognition Using Deep Learning Network Algorithms

A human hand has various functions and lots of flexibility to adapt to daily activities humans perform. There are patients that lost their normal hand function because of problems like physical trauma or nervous system damage. To regain the functionality of the upper limbs to the patients, researchers use prostheses with BCI(Brain-computer interface). The purpose of the research is to determine the applicability of new machine learning algorithms on extracting hand motion intentions from EEG.

BCI does not have a direct connection to muscles and nerves and gets signals from the brain. However, using BCI brings disadvantages such as EEG signals are weak signals which include noises and have low accuracy and poor stability. The research uses the CSP(Common Spatial Pattern) algorithm to overcome this problem.[7]

This research uses a spatial filtering method called filter bank common spatial patterns(FBCSP) and combines it with CNN(Convolutional Neural Networks)which uses one-versus-rest(OVR) to accurately perceive the multi-class hand motion EEG.[7] The reason why OVR is used, since the CSP algorithm is a solution for binary classification problems, to achieve N binary classification OVR is used.[7] In short, The researchers proposed a new CNN classifier that combines feature selection and spatial filtering for multi-classification of motor imagining and hand motion EEG signals for 3 hand movements which are hand opening, closing, and resting.

In this research, two different datasets are used. The first one is Upper limb movement decoding from EEG which includes subjects from 22 years old to 40 years old.[7] The second one is EEG data collected from Xi'an Jiaotong University XiaodongZhang's team and the subjects are 23 years old right-handed men. The datasets are collected from a 4*16 channel amplifier and a NEUROSCAN LABS 32 channel EEG amplifier. Subjects were asked to avoid eye movement during data collection and they were asked to rest between each test. Tests are applied for 2 different hand motion patterns of a hand which are hand opening, closing, resting.[7] Then the data is preprocessed. The stages of preprocessing are as follows. Correcting the sampling frequency of the signal, performing FIR filter processing on each signal, and re-referencing all EEG channels. (The sample size is increased to provide large datasets for deep learning). 70% of the data is used to train the system and fit the CNN model and the remaining 30% of the data is used for testing purposes.[7]

EEG-Based BCI System to Detect Fingers Movements

In the study, a system was proposed to control prosthetic fingers with the adoption of BCI to help people disabled people suffering from motor mobility impairment. The study focused on the multi-class classification problem.[8]

Previous EEG studies extracted from the same area of the brain(same electrodes) for finger movements at a hand But for the same motor imagery task, the activity of the brain in the left and right hemispheres is different So the study states that all electrodes need to be considered.[8] This led them to use a model (which is statistical) to identify the imagery task of each finger for each subject.[8] During the study, they focused on five motor imagery tasks which are pinky finger, ring finger, middle finger, index finger, and the thumb. For the experiments a g.HIamp from g.tec, an 80-channel amplifier was used for the creation of the dataset. The data was processed by removing artifacts; it helped to increase the signal-to-noise ratio of the EEG signals.[8] For feature extraction of the finger movements, the CSP algorithm was applied. For the experiment, they developed an SPI to guide subjects in each session.[8]

After that, a filter block was used.to eliminate artifacts. To allow subtraction of Signals in each electrode for each time point from the average signal and allow calculation of average signal at all electrodes, a technique called the average common average referencing technique was applied. This technique helps to discriminate between the positive and negative peaks in the EEG signals.[8] It is known that for the selection of the relevant electrodes, the previous studies extract features from a predefined set of electrodes to detect the movements.[8] Unlike them, The study provides a method to analyze the brain activity during movements of the fingers. For the classification of movements, the study denotes a one-class classifier for every finger movement of a corresponding finger. This approach allows the model to identify (predict) movements of each finger at the same time. In short, for feature extraction CSP spatial filters are used and for classification, the LDA algorithm is used. [8]

Note: The researchers tested many classifiers such as SVM, Logistic Regression, Gaussian Naive Bayes, and LDA. They decided to go with LDA due to its efficiency and accuracy compared to other classifiers.[8]

Artefacts Removal of EEG Signals with Wavelet Denoising

The EEG signals usually have many noise signals that are coming from muscle activation(EMG). The noise amplitude can be huge relative to the amplitude of the EEG signals. In the paper, a wavelet denoising method is proposed for noise(EMG) removal from EEG signals. The result of the paper says that noise parts of EEG signals can be fairly removed.

Eight male subjects joined the experiment. They were aged 20-22 years old. All of them were thin-haired and had no abnormalities. An epoch finishes in 80 seconds. It starts with silence (first 18 seconds), and then 2 seconds for preparation after that the subject blinks periodically to the end.

The time complexity was linear and the wavelet coefficients were quite small numbers that were close to zero. These properties are suitable for data compression. The input data is filtered with a low pass filter and the high pass filter. The result of the lowpass filter is called approximation and the result of the high pass filter is called detail. The experiment says that denoising can be done by decomposition and reconstruction using approximations and details.

For denoising operation, the form of the signal threshold wavelet should be downloaded. To discrete wavelet transform, input coefficients should be used. According to the paper, wavelet transform can remove noise and/or other undesirable signals, and desired signals can be obtained after an inverse wavelet operation.**[9]**

3.2. Solution Method

The solution includes dataset creation, data cleaning, feature extraction, model training and including the model in the systems. These are explained below.

Data creation: Since the data obtained from two different devices can differ (Their sampling rate, channel number, etc. might be differ), an additional UI program was developed to create our own dataset (It would not be easy to find a dataset that serves our purpose and compatible with the device we have). In that way, the prediction input will be compatible with the model and since the same device is used in the prediction, the results might be better.

Data Cleaning: Since the data obtained from the brain via BCI is also a signal, To cleanse the signal from noise and artifacts, digital signal processing techniques need to be used. In the project, a bandpass and a bandstop filter were used. To clean the data, a bandpass filter and denoising were applied to the signal. Then, after the feature extraction, an outlier detection technique was applied for data cleaning. After outlier detection and elimination, As a result of this, there was an imbalance between the number of supports of the labels in the training. To overcome this problem, an oversampling method was used and the support numbers were equalized.

Feature Extraction: To increase the number of the features, the band powers of each channel were obtained. Eight different bands were chosen. Also, some of the bands contain some crucial information related to the activity performed. The band names and their frequency range are as follows.

- Theta Low (4.0, 5.5)
- Theta High (5.5, 7)
- Alpha Low (7, 10)
- Alpha High (10, 13)
- Beta Low (14, 22)
- Beta High (22, 30)
- Gamma Low (31, 38)
- Gamma High (38, 45)

With that way 8 features were created for each available feature and the number of features were increased from 16 to 128.

Machine learning: To train the system, machine learning methods are going to be used. We referred to some of them in the literature research section. To give an example of the algorithms used: Support Vector Machines (SVM), Logistic Regression, Linear Discriminant Analysis, Random Forest Classifier, Gradient Boosting Classifier (GB), Multinomial Naive Bayes, Decision Tree Classifier, KNeighborsClassifier (KNN), and Voting Classifier methods.

Grid Search: The mentioned machine learning algorithms take different parameters and the results and accuracy also depend on this. To find the optimal parameter, grid search is applied for each model. Grid Search allowed us to find the most performant parameters for each machine learning method

4. ANALYSIS AND MODELING

The model consists of three devices and their interaction between them and each device has a particular role associated with its purpose. The BCI device is responsible for converting brain signals into numerical data, The PiCar is the end device that is responsible for gathering the classified direction. The computer is the main part of the project. All work done will be working on the computer. It is responsible for most of the functionalities of the project. The majority of the work that will be done is going to be related to the computer.

The project will consist of two separate parts. The first part consists of an application to obtain data and the scripts to train a model to predict the direction. The second part of the project consists of replacing the Rule-based model that works with the accelerometer with the model trained. The trained model in the first part will be used to predict the direction.

In the model collection part, different visual stimulants will be used for gestures and MI. But since the general course of the transactions will be the same, there is no need to explain it differently. The same ML algorithms will be used in the model creation processes required for the development of these systems, and only the parameters found to be suitable for them will be changed with GridSearch. The benefit of this is that it will be possible to achieve higher F1 score and accuracy for both different collected data.

4.1. System Factors

One of the most important external factors of the system is a proper internet connection. There is an obligation to be connected to the same internet. Different connection methods may be provided in the future. If the connection is not consistent enough, some problems may occur.
The host computer both needs to collect data from the dongle instantly and also needs to process that data. It is important that these parts work without errors.

4.2. How System Works

First of all, it should be said that the system has more than one part. While it is very important that they work together and correctly, it is the basic element that makes the project work.

It is very important that the OpenBCI EEG Headset works correctly and that the data can be instantly transferred to the computer with the help of the dongle. Thus, that data can be reworked with certain methods and filters, and the data can be collected in a noise-free manner.

After that, the number of features was increased before that data was given to machine learning algorithms. For this, the collected data has been kept in different files with different frequency values. These are alpha, beta, gamma, and theta, together with their high and low versions, 8 different files are obtained.

The values in certain frequency ranges collected in these files are basically cleaned by outlier detection and elimination before being given to machine learning algorithms, and oversampling is followed in order to eliminate the deficiency in the number of support that will occur after that. In this part, a difference is made for EEG. Before applying these processes, it is possible to create more values such as alpha and beta with the data in hand, with a special method. This method works like this: For example, you have 500 units of data. In order to create an alpha value from this data, you need to round it as 100-100-100-100-100. Thus, you will be able to get 5 different alpha values. However, instead of doing this, more data is created by progressing step by step, that is, between 0-100, between 25-125, and between 50-150. The jump number here, 25, may vary.

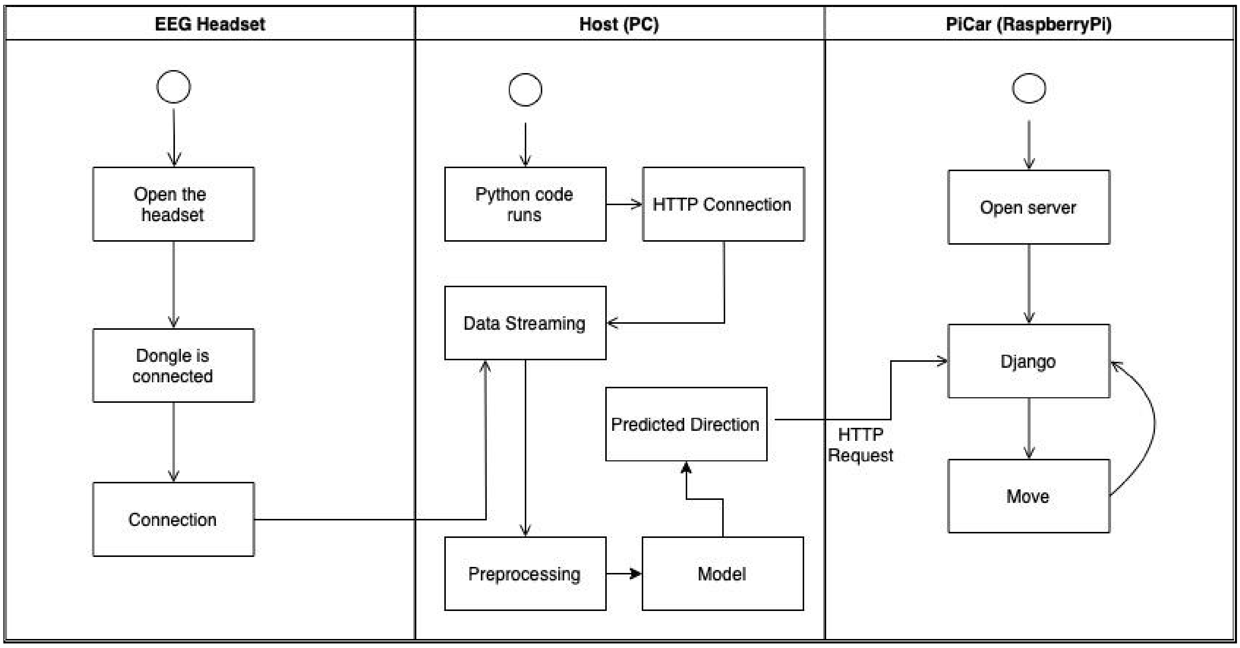
After that, models are developed with the alpha and beta files available. The most suitable parameters of these models were tried to be found with GridSearch. With the obtained parameters, it has been tried to ensure that there is no overfit or underfit, and for this, some parameters have been tried. The model obtained from here is extracted as a file with the library named Pickle.

This model, on the other hand, can make predictions with a live-streaming session. In this part, the data comes from the headset instantly and it is divided into values such as alpha, beta, and predictions are made quickly. This system, which is designed to make 1 prediction in 2.5 seconds, is designed to collect 5 pieces of 500ms data and take their mode and choose whichever direction is the most. The results obtained are sent to PiCar with an HTTP request and it is possible to observe it.

4.3. Modeling

How the system works, in general, has been explained above, and it can be said that similar steps have been taken in the modeling part.

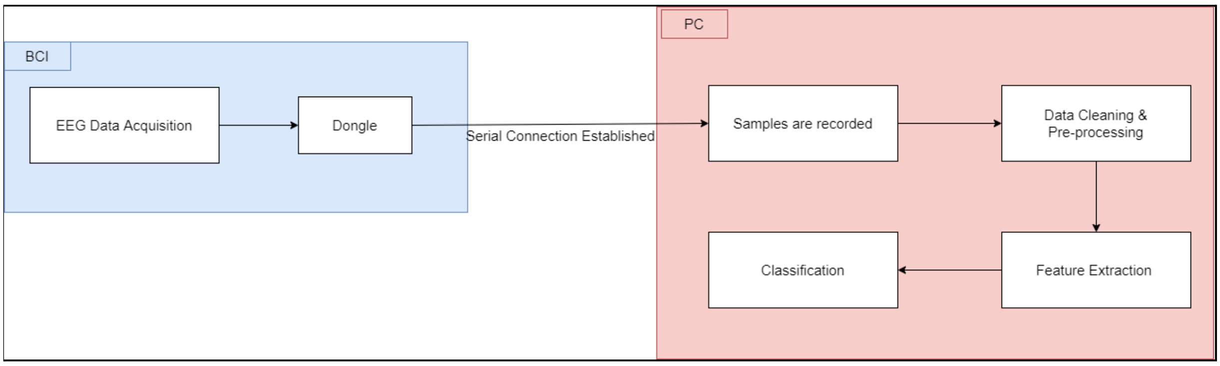
4.3.1. System Architecture



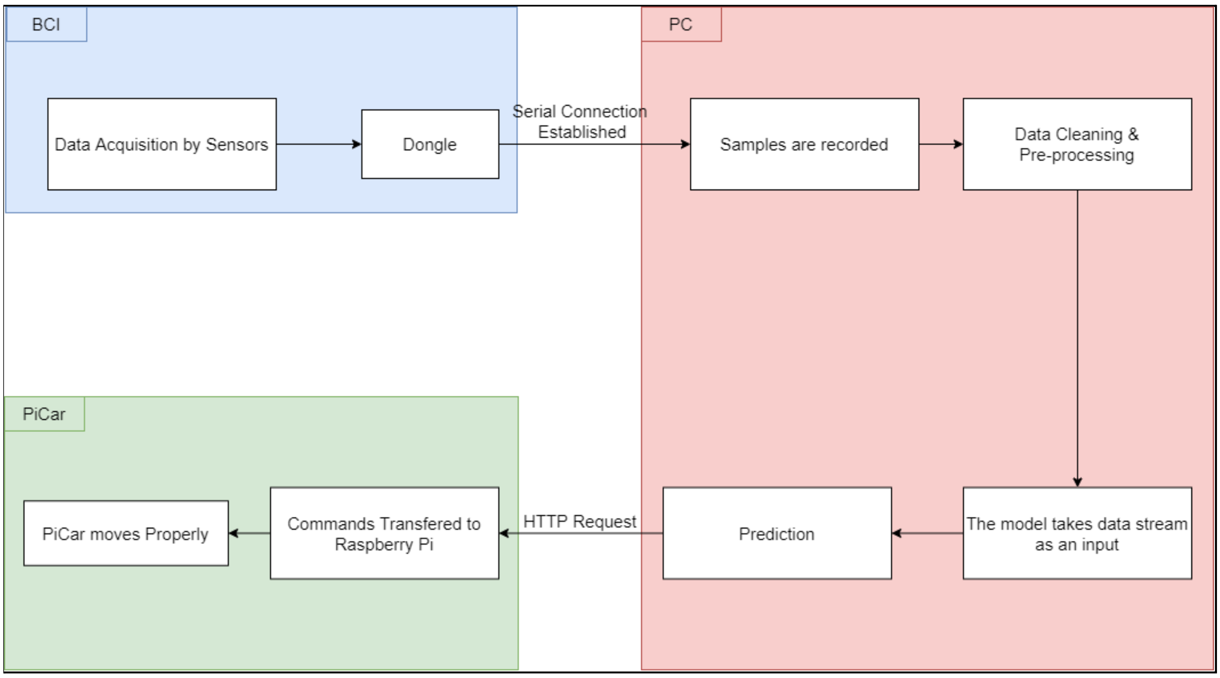
Figure#5: System Architecture

4.3.2. UML (Unified Modeling Language) Diagrams

Below UML diagrams are showing the training and prediction part of the project.



Figure#6: Training UML



Figure#7: Prediction UML

5. DESIGN, IMPLEMENTATION AND TESTING

5.1. Design

From the design perspective, the project consists of three devices that are in interaction between them. These devices are BCI, computer, and the PiCar. Their design is shaped around the tasks that they are responsible for. BCI and PiCar have relatively narrow perspectives. The computer includes more tasks than the BCI and PiCar has.

Brain-Computer Interface (Cython/Daisey module)

Gather the electrical activity around the different nodes from a human's head using dry electrodes and turn them into numerical quantities. Then, sends them into a computer via a dongle in the sampling rate that it is capable of.

Computer

In the Computer part, the system consists of consecutive steps. While designing the system, our approach was to design a pipeline that performs the steps in the desired order. Each step in the pipeline can be considered as a subsystem that is designed in order to achieve the task that it is responsible for. These steps are given below.

- **Data obtain and filtering system:** This sub-system is responsible for communicating with the BCI and obtaining the data. It is supported by a user interface that directs the people. While directing, It also gathers the data from BCI in parallel and stores it in a buffer. After that, the signals are filtered using bandpass and bandstop filters in order to clean the signal.

- **Feature extraction/Preprocessing:** This sub-system applies preprocessing methods to the data and creates extra features (signals in different bands). The dataset created in the previous step is used.

- **Machine Learning:** After the dataset passes through the noise cleaning, feature extraction, and preprocessing steps. The data dataset is ready to be used in a machine learning model. This subsystem trains the models and saves them for future use

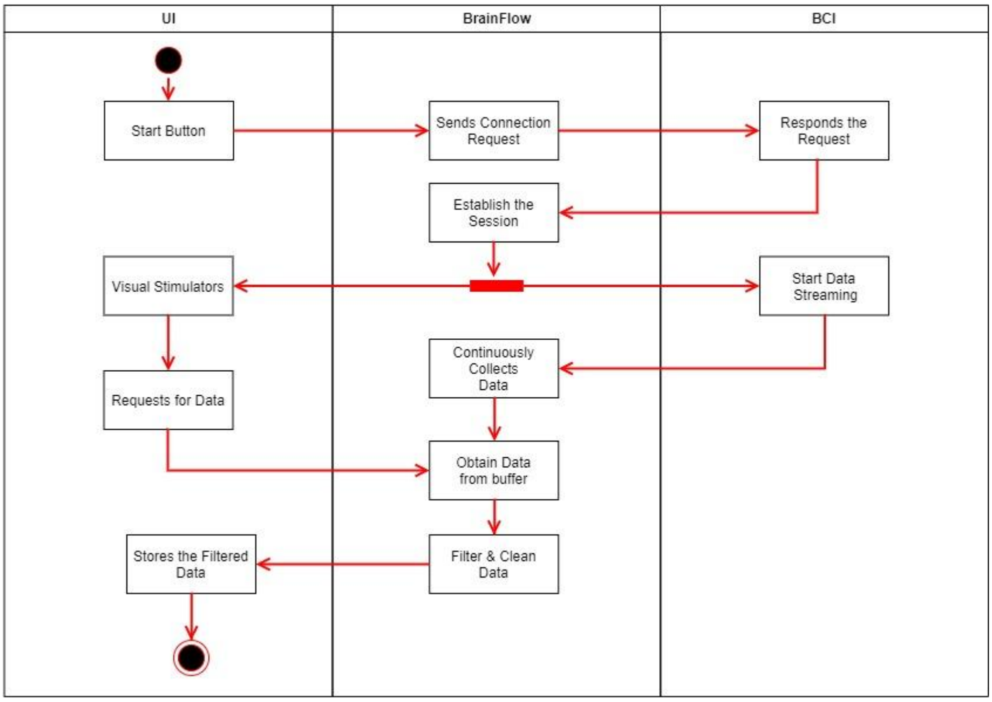
- **Prediction:** This subsystem obtains the data from the user every 2.5 seconds. Here, the models saved in the previous step are loaded and the data obtained within the 2.5 seconds are passed as a parameter to the model. After the machine learning model makes a prediction, the prediction is sent to the PiCar using the HTTP protocol.

PiCar: PiCar performs a movement through the predicted direction. It has a web server available in it (which is written using the Django framework). This web server listens to 8000 port and it performs the actions that it received via an HTTP request from 8000 port.

5.2. Implementation

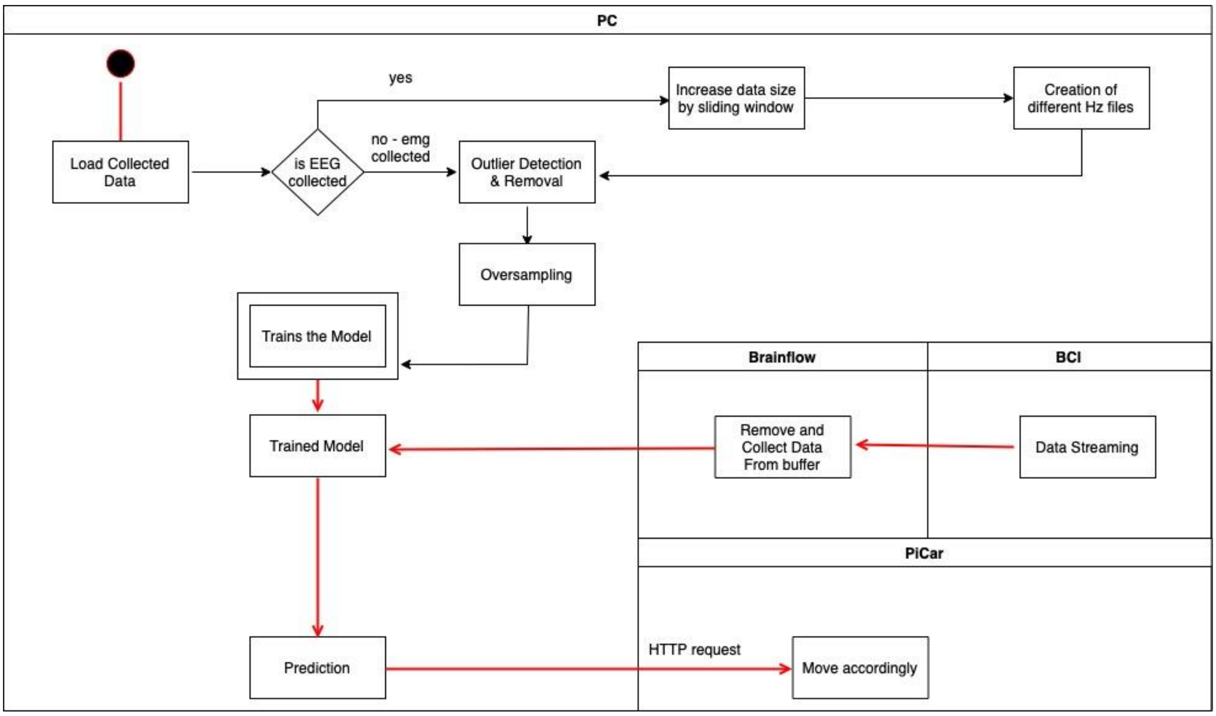
The main purpose of this project was to develop a model with motor imagery (MI) or mental imagery and bring mobility to the devices with that model. However, if the models developed with this method did not have high enough accuracy and F1 scores, you had to have a backup plan. For this, the idea of developing a model with the observation of motion, specifically gestures emerged as a backup plan. It was possible to quickly switch between plans, thanks to the almost completely generic development of all properties except the way the data was collected so that the results of both plans could be shared and compared.

The overall design is based on how data is collected. For this, we have two different data collection systems and we have integrated them for different plans. The collection logic of the data collected by the MI and gesture methods is almost the same. First of all, both of them come with an interface created with the python Tkinter library, and animated images (gifs) are brought to the screen in different positions. This interface, which has 5 different gifs, keeps all these gifs on the screen for 5 seconds one by one. A blank screen appears for 0.5 seconds when changing directions. The data collected during the directions are taken from the buffer of the EEG headset at the end of every 0.5 seconds and the processes are started to be applied to it. First, the mean or best line of the detrended data is found and this is done for each column separately. This data is drawn to the range of 1-46Hz with the bandpass filter and allows it to work in the meantime. The standard frequency of the European electricity grid, 50Hz, is blocked by a bandstop filter. After that, wavelet_denoising is done. Thus, it is possible to get rid of most of the noise. The data obtained after these processes is divided into parts as follows: alpha low (7 - 10 Hz), alpha high (10 - 13 Hz), beta low (14 - 22 Hz), beta high (22 - 30 Hz), gamma low (31 - 38 Hz), gamma high (38 - 45 Hz), theta low (4 - 5.5 Hz), and theta high (5.5 - 7 Hz). All of these are collected in a separate .csv file and all these calculations are made for each channel (16 different channels) and recorded separately. The operations performed here were implemented with the help of the Brainflow library by determining the correct parameters.



Figure#8: Data Collection UML

Different machine learning models were developed using the .csv files collected here. First of all, these data should answer simple but effective questions such as whether they are logical and follow a pattern. Outlier Detection is performed by selecting rbf as the kernel of OneClassSVM. Thus, the locations of useless or corrupted data are detected and these data are removed from the process. After these are completed, we solve the support imbalance of the data (directions) that we destroyed in the Outlier Detection section by performing SMOTE oversampling. Thus, we have the same number of data that we can test and train for each direction. We develop models with Support Vector Machines (SVM), Logistic Regression, Linear Discriminant Analysis, Random Forest Classifier, Gradient Boosting Classifier (XGB), Multinomial Naive Bayes, Decision Tree Classifier, KNeighborsClassifier (KNN), and Voting Classifier methods, respectively. Fine-tuning processes were carried out for all of them separately, and the model was developed in a short time by using the best parameters. With the Pickle library, we take the model with the highest F1 and accuracy and put it in the environment we have developed to make instant predictions. In this environment, data is collected in the same way and goes through the same processes. However, 500ms of data collected and processed here is kept for 1.5 seconds and at the end such an estimate is made. In other words, a direction estimation is made every 1.5 seconds and this estimation is made by looking at the mode of 3 different data. Thus, it is easier to obtain more accurate results and a more appropriate reflex time is given to the human mind.



Figure#9: Training and Prediction UML

5.3. Testing

While testing the models two different methods are used. The first method was train_test_split() function and the second method was splitting the dataset as test and train sets by hand. After the implementation of the first method, we observed that the model developed with EEG was not as good as the accuracy obtained with the first method. So we went to the second method and set the last 2 trials as test sets by hand. We accepted the results from here. To verify which model performs better, a confusion matrix was created for each model. The confusion matrices provided us with different comparison metrics such as Precision, Recall F1 score, and accuracy. Also, for each model, the confusion matrices for both the train and test parts were observed. These matrices were used to verify whether the model is working properly. Also, a heatmap was created for each model. Having a confusion matrix and heat maps helped us to compare the performance of the models.

6. RESULTS

There are different results for both datasets (MI and Gestures). As it is seen below, the mental imagery part isn't that high in terms of accuracy. An average of 40% accuracy was achieved in the Mental Imagery section, and as a result, a usable system could not be obtained. In order to increase this, more data collection, performing this experiment with a device with more channels and sampling rates can cause great positive increases in the results. In addition, for the Gestures part, there is an average of 80%-90% accuracy. It would be more realistic to say that this is a more suitable method for this system.

We observed that the model with the highest accuracy in the mental imagery section is SVM. This model, which was developed without overfitting, unfortunately, did not help us get closer to our previously targeted results. Experiencing the disadvantages of the system, especially during the process, played a major role in making the initial estimates wrong.

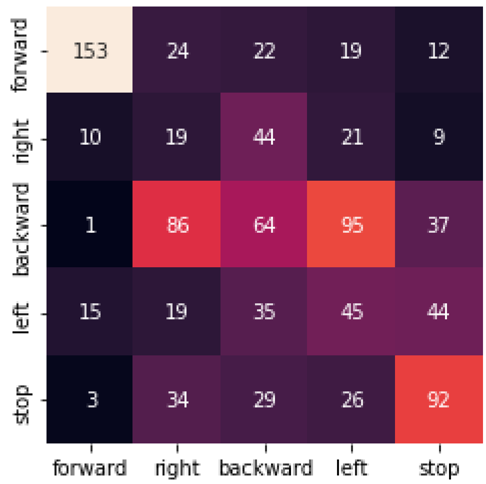
For the Gestures part, we were able to achieve high results. We were able to reach an accuracy of 88%, exceeding our target of 70%, with the Logistic Regression model. Although there are other algorithms with similar ratios, it is possible to say that it works well enough.

MENTAL IMAGERY PART

SVM

	precision	recall	f1-score	support
forward	0.67	0.84	0.74	182
right	0.18	0.10	0.13	182
backward	0.23	0.33	0.27	194
left	0.28	0.22	0.25	206
stop	0.50	0.47	0.49	194
accuracy			0.41	958
macro avg	0.40	0.42	0.40	958
weighted avg	0.40	0.41	0.40	958
Best params: {'svc__C': 0.2, 'svc__degree': 2, 'svc__gamma': 0.0001, 'svc__kernel': 'linear'}				

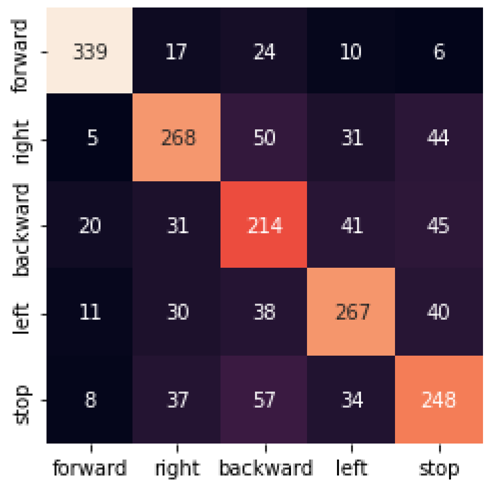
Table 2. SVM-EEG Test Prediction Result



Figure#10: SVM-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.86	0.89	0.87	383
right	0.67	0.70	0.69	383
backward	0.61	0.56	0.58	383
left	0.69	0.70	0.69	383
stop	0.65	0.65	0.65	383
accuracy			0.70	1915
macro avg	0.70	0.70	0.70	1915
weighted avg	0.70	0.70	0.70	1915
Best params: {'svc__C': 0.2, 'svc__degree': 2, 'svc__gamma': 0.0001, 'svc__kernel': 'linear'}				

Table 3. SVM-EEG Train Prediction Result

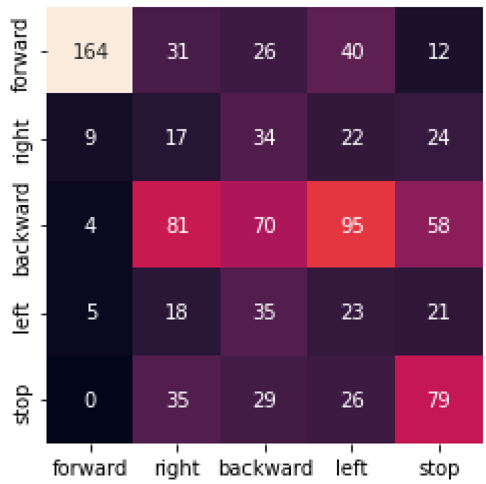


Figure#11: SVM-EEG Heat Map of Train Prediction

Logistic Regression

	precision	recall	f1-score	support
forward	0.60	0.90	0.72	182
right	0.16	0.09	0.12	182
backward	0.23	0.36	0.28	194
left	0.23	0.11	0.15	206
stop	0.47	0.41	0.44	194
accuracy			0.37	958
macro avg	0.34	0.37	0.34	958
weighted avg	0.33	0.37	0.34	958
Best params: {'C': 2, 'intercept_scaling': 1, 'l1_ratio': 2, 'penalty': 'l1', 'solver': 'liblinear'}				

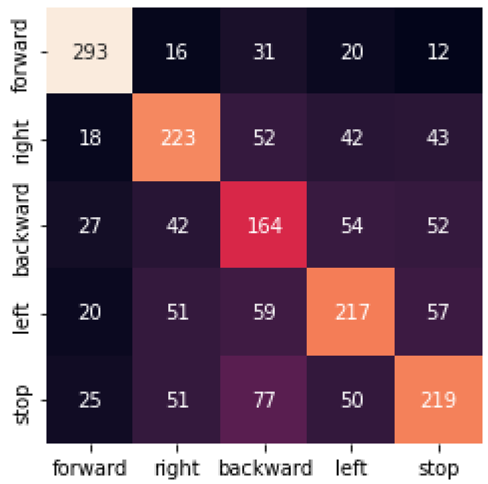
Table 4. LR-EEG Test Prediction Result



Figure#12: LR-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.79	0.77	0.78	383
right	0.59	0.58	0.59	383
backward	0.48	0.43	0.45	383
left	0.54	0.57	0.55	383
stop	0.52	0.57	0.54	383
accuracy			0.58	1915
macro avg	0.58	0.58	0.58	1915
weighted avg	0.58	0.58	0.58	1915
Best params: {'C': 2, 'intercept_scaling': 1, 'l1_ratio': 2, 'penalty': 'l1', 'solver': 'liblinear'}				

Table 5. LR-EEG Train Prediction Result

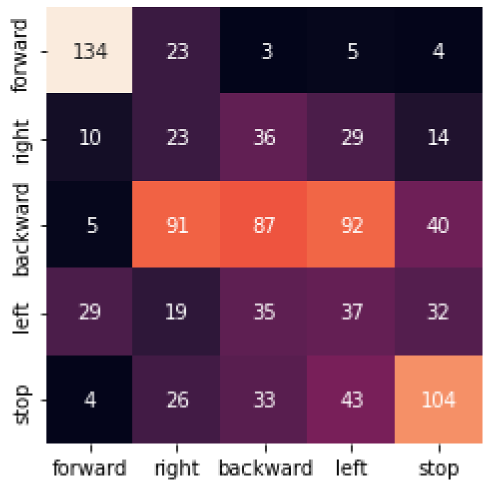


Figure#13: LR-EEG Heat Map of Train Prediction

Linear Discriminant Analysis

	precision	recall	f1-score	support
forward	0.79	0.74	0.76	182
right	0.21	0.13	0.16	182
backward	0.28	0.45	0.34	194
left	0.24	0.18	0.21	206
stop	0.50	0.54	0.51	194
accuracy			0.40	958
macro avg	0.40	0.41	0.40	958
weighted avg	0.40	0.40	0.39	958
Best params: {'n_components': 1, 'shrinkage': None, 'solver': 'lsqr', 'store_covariance':True}				

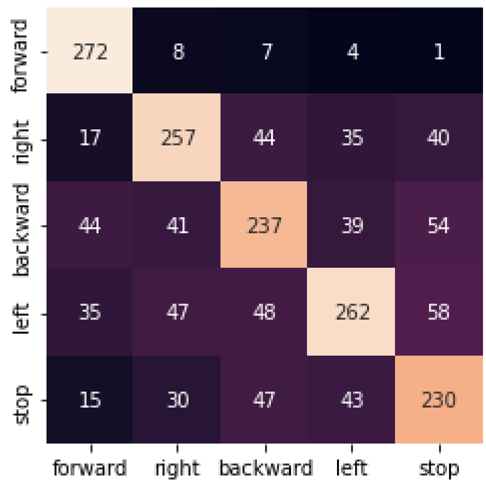
Table 6. LDA-EEG Test Prediction Result



Figure#14: LDA-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.93	0.71	0.81	383
right	0.65	0.67	0.66	383
backward	0.57	0.62	0.59	383
left	0.58	0.68	0.63	383
stop	0.63	0.60	0.61	383
accuracy			0.66	1915
macro avg	0.67	0.66	0.66	1915
weighted avg	0.67	0.66	0.66	1915
Best params:{'n_components': 1, 'shrinkage': None, 'solver': 'lsqr', 'store_covariance':True}				

Table 7. LDA-EEG Train Prediction Result

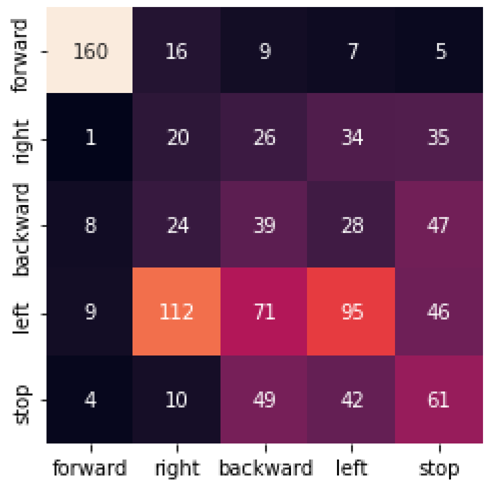


Figure#15: LDA-EEG Heat Map of Train Prediction

Random Forest Classification

	precision	recall	f1-score	support
forward	0.81	0.88	0.84	182
right	0.17	0.11	0.13	182
backward	0.27	0.20	0.23	194
left	0.29	0.46	0.35	206
stop	0.37	0.31	0.34	194
accuracy			0.39	958
macro avg	0.38	0.39	0.38	958
weighted avg	0.38	0.39	0.38	958
Best params: {'criterion': 'gini', 'max_features': 'auto', 'n_estimators': 200, 'verbose': 0}				

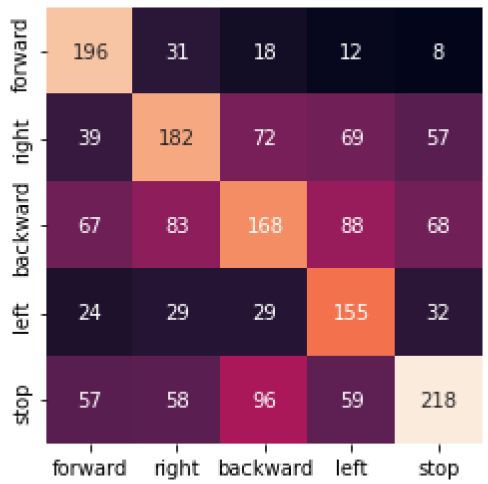
Table 8. RFC-EEG Test Prediction Result



Figure#16: RFC-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.74	0.51	0.60	383
right	0.43	0.48	0.45	383
backward	0.35	0.44	0.39	383
left	0.58	0.40	0.48	383
stop	0.45	0.57	0.50	383
accuracy			0.48	1915
macro avg	0.51	0.48	0.49	1915
weighted avg	0.51	0.48	0.49	1915
Best params: {'criterion': 'gini', 'max_features': 'auto', 'n_estimators': 200, 'verbose': 0}				

Table 9. RFC-EEG Train Prediction Result

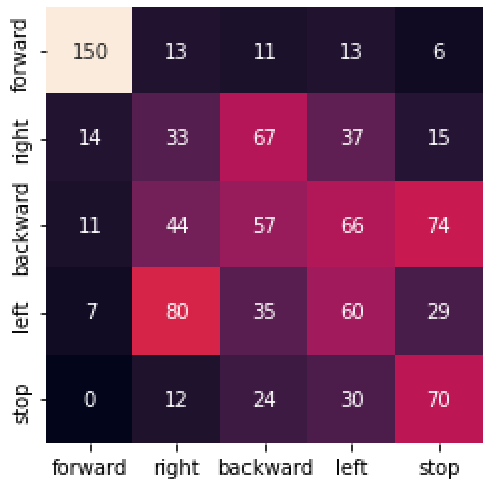


Figure#17: RFC-EEG Heat Map of Train Prediction

Gradient Boosting Classification

	precision	recall	f1-score	support
forward	0.78	0.82	0.80	182
right	0.20	0.18	0.19	182
backward	0.23	0.29	0.26	194
left	0.28	0.29	0.29	206
stop	0.51	0.36	0.42	194
accuracy			0.39	958
macro avg	0.40	0.39	0.39	958
weighted avg	0.40	0.39	0.39	958
Best params: {'learning_rate': 0.15, 'loss': 'deviance', 'n_estimators': 120, 'verbose': 0}				

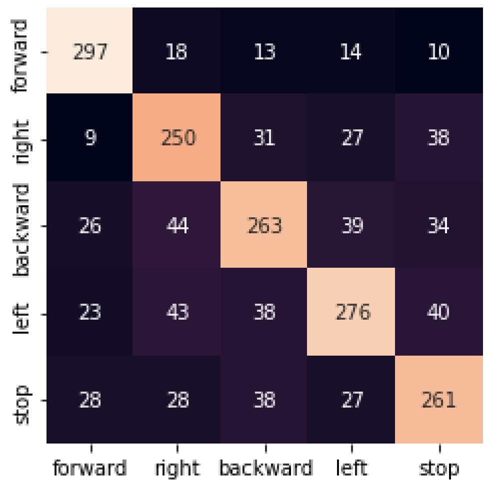
Table 10. GBC-EEG Test Prediction Result



Figure#18: GBC-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.84	0.78	0.81	383
right	0.70	0.65	0.68	383
backward	0.65	0.69	0.67	383
left	0.66	0.72	0.69	383
stop	0.68	0.68	0.68	383
accuracy			0.70	1915
macro avg	0.71	0.70	0.70	1915
weighted avg	0.71	0.70	0.70	1915
Best params: {'learning_rate': 0.15, 'loss': 'deviance', 'n_estimators': 120, 'verbose': 0}				

Table 11. GBC-EEG Train Prediction Result

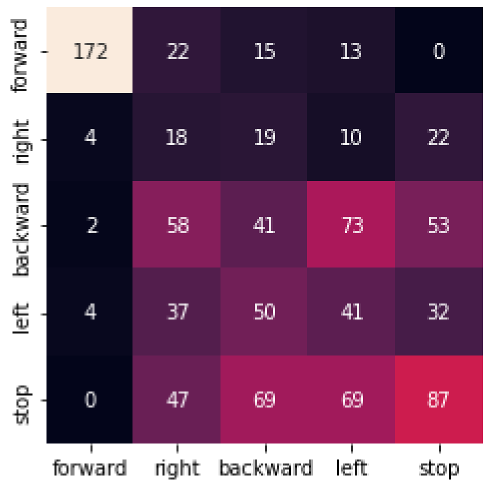


Figure#19: GBC-EEG Heat Map of Train Prediction

Multinomial Naive Bayesian

	precision	recall	f1-score	support
forward	0.77	0.95	0.85	182
right	0.25	0.10	0.14	182
backward	0.18	0.21	0.19	194
left	0.25	0.20	0.22	206
stop	0.32	0.45	0.37	194
accuracy			0.37	958
macro avg	0.35	0.38	0.36	958
weighted avg	0.35	0.37	0.35	958
Best params: {'alpha': 1.0}				

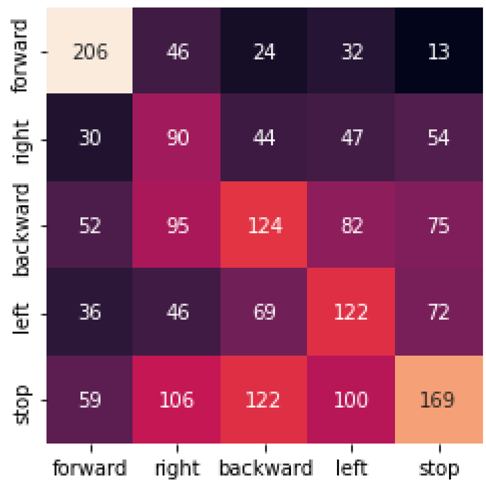
Table 12. MNB-EEG Test Prediction Result



Figure#20: MNB-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.64	0.65	0.59	383
right	0.34	0.23	0.28	383
backward	0.29	0.32	0.31	383
left	0.35	0.32	0.34	383
stop	0.30	0.44	0.36	383
accuracy			0.37	1915
macro avg	0.39	0.37	0.37	1915
weighted avg	0.39	0.37	0.37	1915
Best params: {'alpha': 1.0}				

Table 13. MNB-EEG Train Prediction Result

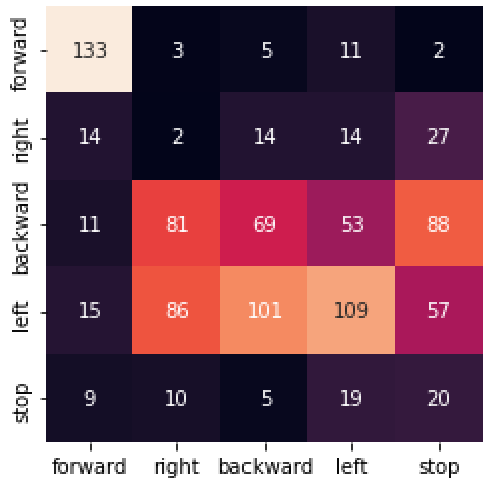


Figure#21: MNB-EEG Heat Map of Train Prediction

Decision Tree Classification

	precision	recall	f1-score	support
forward	0.86	0.73	0.79	182
right	0.03	0.01	0.02	182
backward	0.23	0.36	0.28	194
left	0.30	0.53	0.38	206
stop	0.32	0.10	0.16	194
accuracy			0.35	958
macro avg	0.35	0.35	0.32	958
weighted avg	0.34	0.35	0.32	958
Best params: {'criterion': 'gini', 'max_depth': 10, 'min_samples_split': 2, 'splitter': 'random'}				

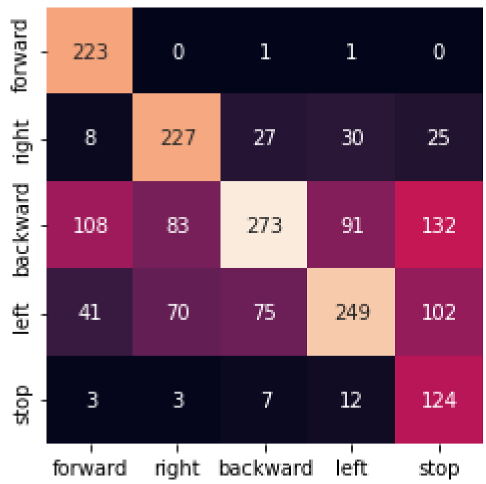
Table 14. DTC-EEG Test Prediction Result



Figure#22: DTC-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.99	0.58	0.73	383
right	0.72	0.59	0.65	383
backward	0.40	0.71	0.51	383
left	0.46	0.65	0.54	383
stop	0.83	0.32	0.47	383
accuracy			0.57	1915
macro avg	0.68	0.57	0.58	1915
weighted avg	0.68	0.57	0.58	1915
Best params: {'criterion': 'gini', 'max_depth': 10, 'min_samples_split': 2, 'splitter': 'random'}				

Table 15. DTC-EEG Train Prediction Result

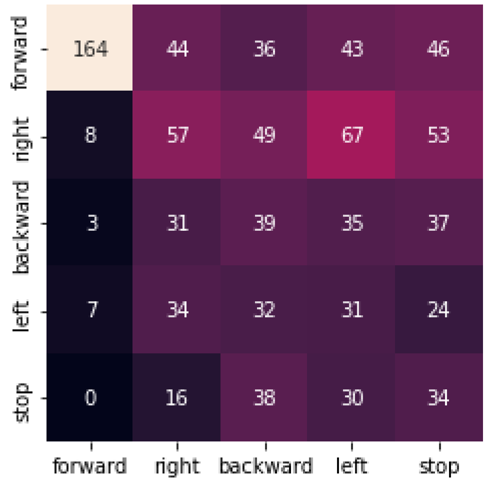


Figure#23: DTC-EEG Heat Map of Train Prediction

K-Nearest Neighbor Classification

	precision	recall	f1-score	support
forward	0.49	0.90	0.64	182
right	0.24	0.31	0.27	182
backward	0.27	0.20	0.23	194
left	0.24	0.15	0.19	206
stop	0.29	0.18	0.22	194
accuracy			0.34	958
macro avg	0.31	0.35	0.31	958
weighted avg	0.30	0.34	0.30	958
Best params: {'algorithm': 'ball_tree', 'n_neighbors': 4, 'weights': 'uniform'}				

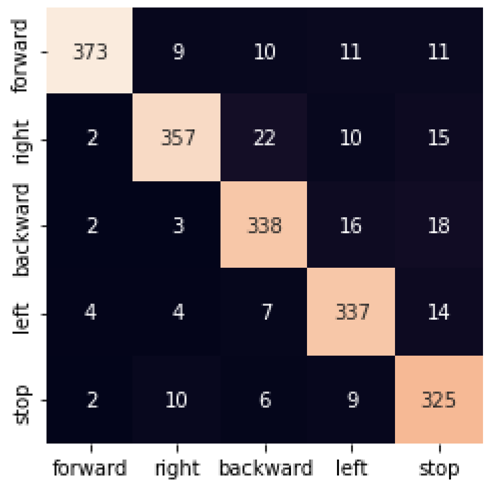
Table 16. KNN-EEG Test Prediction Result



Figure#24: KNN-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.90	0.97	0.94	383
right	0.88	0.93	0.90	383
backward	0.90	0.88	0.89	383
left	0.92	0.88	0.90	383
stop	0.92	0.85	0.88	383
accuracy			0.90	1915
macro avg	0.90	0.90	0.90	1915
weighted avg	0.90	0.90	0.90	1915
Best params: {'algorithm': 'ball_tree', 'n_neighbors': 4, 'weights': 'uniform'}				

Table 17. KNN-EEG Train Prediction Result

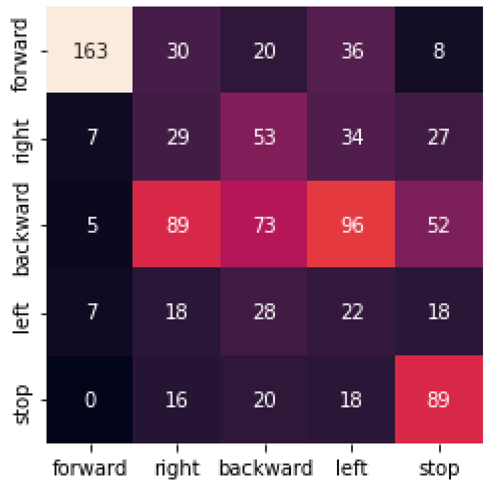


Figure#25: KNN-EEG Heat Map of Train Prediction

Voting Classification

	precision	recall	f1-score	support
forward	0.63	0.90	0.74	182
right	0.19	0.16	0.17	182
backward	0.23	0.38	0.29	194
left	0.24	0.11	0.15	206
stop	0.62	0.46	0.53	194
accuracy			0.39	958
macro avg	0.38	0.40	0.38	958
weighted avg	0.38	0.39	0.37	958
Used Methods: Linear Discriminant Analysis, Gradient Boosting Classification, Logistic regression				

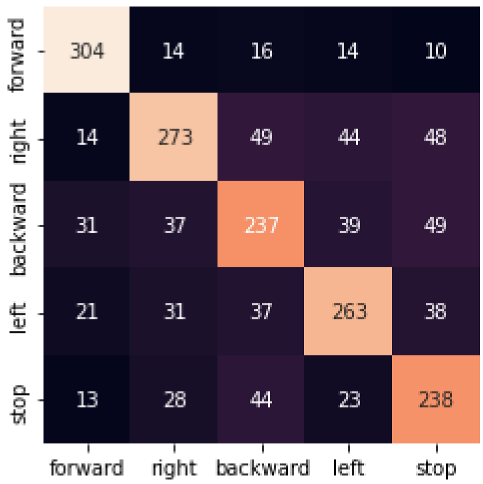
Table 18. VC-EEG Test Prediction Result



Figure#26: VC-EEG Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.85	0.79	0.82	383
right	0.64	0.71	0.67	383
backward	0.60	0.62	0.61	383
left	0.67	0.69	0.68	383
stop	0.69	0.62	0.65	383
accuracy			0.69	1915
macro avg	0.69	0.69	0.69	1915
weighted avg	0.69	0.69	0.69	1915
Used Methods: Linear Discriminant Analysis, Gradient Boosting Classification, Logistic regression				

Table 19. VC-EEG Train Prediction Result



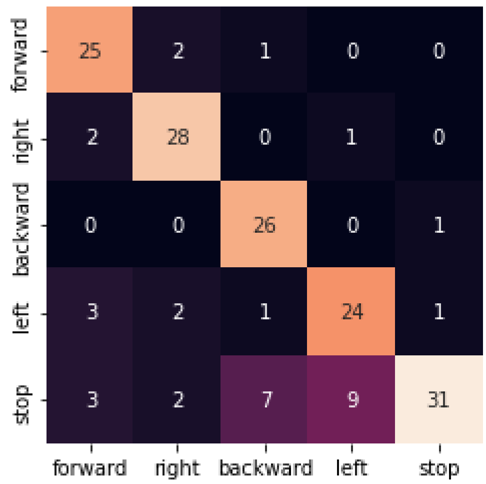
Figure#27: VC-EEG Heat Map of Train Prediction

GESTURES PART

SVM

	precision	recall	f1-score	support
forward	0.89	0.76	0.82	33
right	0.90	0.82	0.86	34
backward	0.96	0.74	0.84	35
left	0.77	0.71	0.74	34
stop	0.60	0.94	0.73	33
accuracy			0.79	169
macro avg	0.83	0.79	0.80	169
weighted avg	0.83	0.79	0.80	169
Best params: {'svc__C': 0.2, 'svc__degree': 2, 'svc__gamma': 0.0001, 'svc__kernel': 'linear'}				

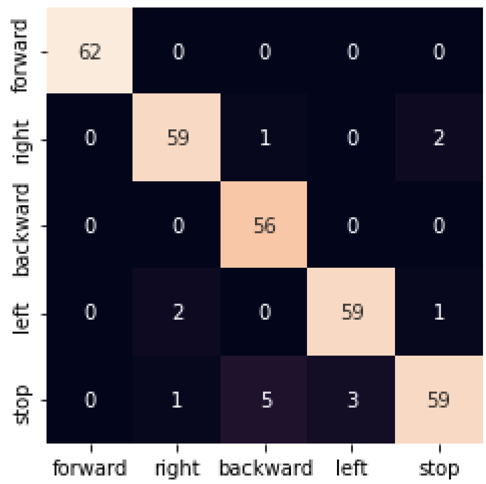
Table 20. SVM-Gestures Test Prediction Result



Figure#28: SVM-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	1.0	1.0	1.0	62
right	0.95	0.95	0.95	62
backward	1.0	0.90	0.95	62
left	0.95	0.95	0.95	62
stop	0.87	0.95	0.91	62
accuracy			0.95	310
macro avg	0.95	0.95	0.95	310
weighted avg	0.95	0.95	0.95	310
Best params: {'svc__C': 0.2, 'svc__degree': 2, 'svc__gamma': 0.0001, 'svc__kernel': 'linear'}				

Table 21. SVM-Gestures Train Prediction Result

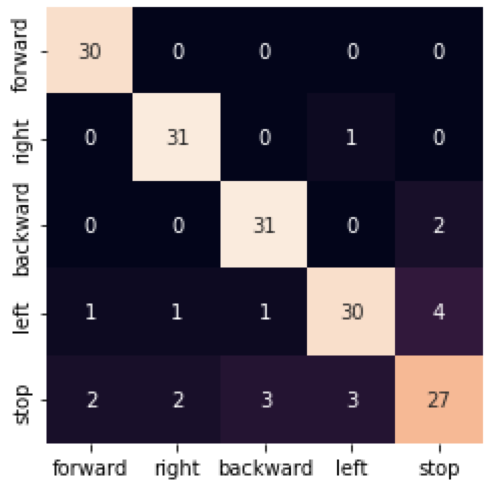


Figure#29: SVM-Gestures Heat Map of Train Prediction

Logistic Regression

	precision	recall	f1-score	support
forward	1.0	0.91	0.95	33
right	0.97	0.91	0.94	34
backward	0.94	0.89	0.91	35
left	0.81	0.88	0.85	34
stop	0.73	0.82	0.77	33
accuracy			0.88	169
macro avg	0.89	0.88	0.88	169
weighted avg	0.89	0.88	0.88	169
Best params: {'C': 1, 'intercept_scaling': 1, 'l1_ratio': 1, 'penalty': 'l2', 'solver': 'sag'}				

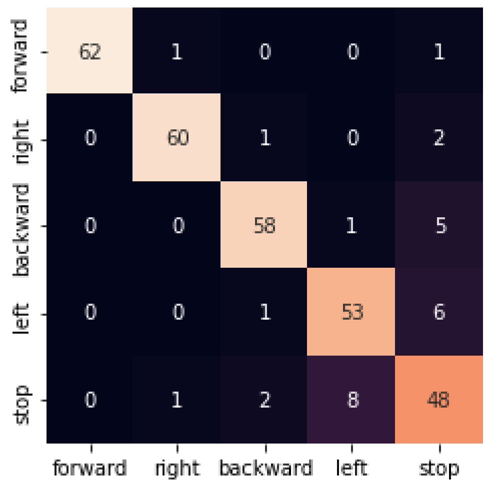
Table 22. LR-Gestures Test Prediction Result



Figure#30: LR-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.97	1.0	0.98	62
right	0.95	0.97	0.96	62
backward	0.94	0.94	0.92	62
left	0.88	0.85	0.87	62
stop	0.81	0.77	0.79	62
accuracy			0.91	310
macro avg	0.90	0.91	0.91	310
weighted avg	0.90	0.91	0.91	310
Best params: {'C': 1, 'intercept_scaling': 1, 'l1_ratio': 1, 'penalty': 'l2', 'solver': 'sag'}				

Table 23. LR-Gestures Train Prediction Result

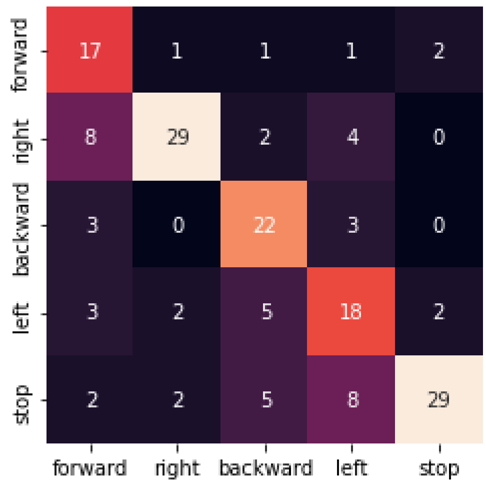


Figure#31: LR-Gestures Heat Map of Train Prediction

Linear Discriminant Analysis

	precision	recall	f1-score	support
forward	0.77	0.52	0.62	33
right	0.67	0.85	0.75	34
backward	0.79	0.63	0.70	35
left	0.60	0.53	0.56	34
stop	0.63	0.88	0.73	33
accuracy			0.68	169
macro avg	0.69	0.68	0.67	169
weighted avg	0.69	0.68	0.67	169
Best params: {'n_components': 1, 'shrinkage': None, 'solver': 'eigen', 'store_covariance': True}				

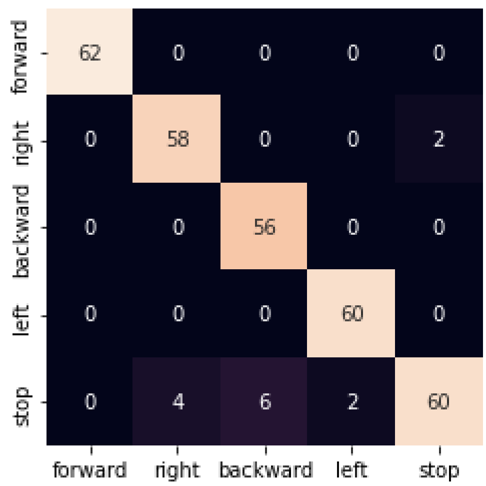
Table 24. LDA-Gestures Test Prediction Result



Figure#32: LDA-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	1.0	1.0	1.0	62
right	0.97	0.94	0.95	62
backward	1.0	0.90	0.95	62
left	1.0	0.97	0.98	62
stop	0.83	0.97	0.90	62
accuracy			0.95	310
macro avg	0.96	0.95	0.96	310
weighted avg	0.96	0.95	0.96	310
Best params: {'n_components': 1, 'shrinkage': None, 'solver': 'eigen', 'store_covariance': True}				

Table 25. LDA-Gestures Train Prediction Result

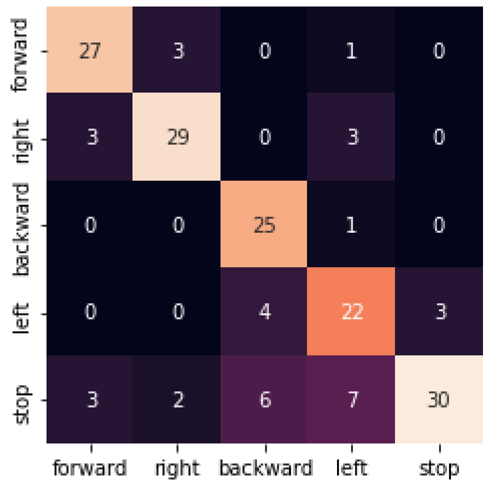


Figure#33: LDA-Gestures Heat Map of Train Prediction

Random Forest Classification

	precision	recall	f1-score	support
forward	0.87	0.82	0.84	33
right	0.83	0.85	0.84	34
backward	0.96	0.71	0.82	35
left	0.76	0.65	0.70	34
stop	0.62	0.91	0.74	33
accuracy			0.79	169
macro avg	0.81	0.79	0.79	169
weighted avg	0.81	0.79	0.79	169
Best params: {'criterion': 'gini', 'max_features': 'auto', 'n_estimators': 100, 'verbose': 0}				

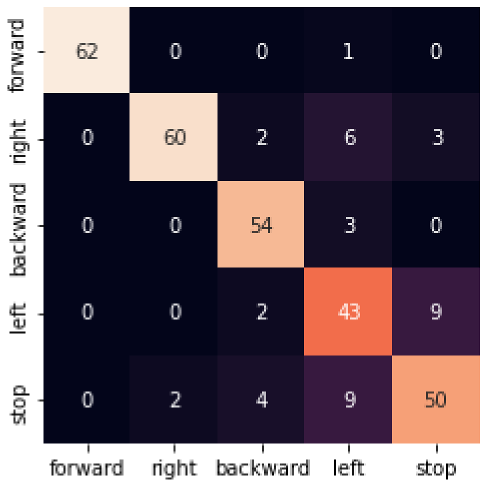
Table 26. RFC-Gestures Test Prediction Result



Figure#34: RFC-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.98	1.0	0.99	62
right	0.85	0.97	0.90	62
backward	0.95	0.87	0.91	62
left	0.80	0.69	0.74	62
stop	0.77	0.81	0.79	62
accuracy			0.87	310
macro avg	0.87	0.87	0.87	310
weighted avg	0.87	0.87	0.87	310
Best params: {'criterion': 'gini', 'max_features': 'auto', 'n_estimators': 100, 'verbose': 0}				

Table 27. RFC-Gestures Train Prediction Result

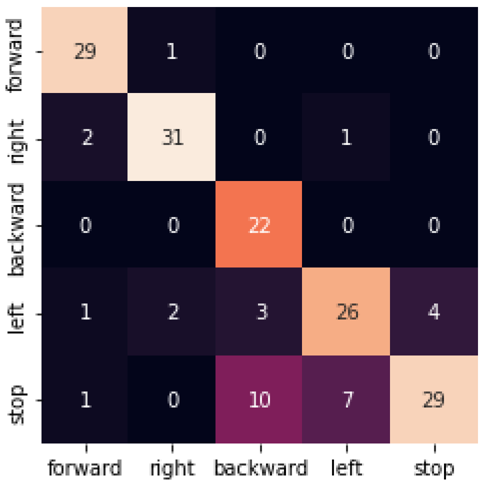


Figure#35: RFC-Gestures Heat Map of Train Prediction

Gradient Boosting Classification

	precision	recall	f1-score	support
forward	0.97	0.88	0.92	33
right	0.91	0.91	0.91	34
backward	1.0	0.63	0.77	35
left	0.72	0.76	0.74	34
stop	0.62	0.88	0.73	33
accuracy			0.81	169
macro avg	0.84	0.81	0.81	169
weighted avg	0.85	0.81	0.81	169
Best params: {'learning_rate': 0.03, 'loss': 'deviance', 'n_estimators': 80, 'verbose': 0}				

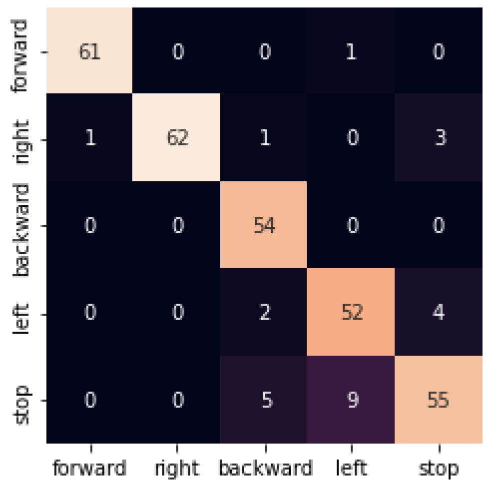
Table 28. GBC-Gestures Test Prediction Result



Figure#36: GBC-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.98	0.98	0.98	62
right	0.93	1.0	0.96	62
backward	1.0	0.87	0.93	62
left	0.90	0.84	0.87	62
stop	0.80	0.89	0.84	62
accuracy			0.92	310
macro avg	0.92	0.92	0.92	310
weighted avg	0.92	0.92	0.92	310
Best params: {'learning_rate': 0.03, 'loss': 'deviance', 'n_estimators': 80, 'verbose': 0}				

Table 29. GBC-Gestures Train Prediction Result

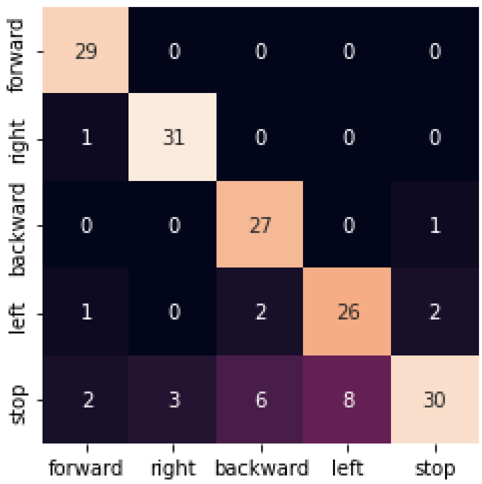


Figure#37: GBC-Gestures Heat Map of Train Prediction

Multinomial Naive Bayesian

	precision	recall	f1-score	support
forward	1.0	0.88	0.94	33
right	0.97	0.91	0.94	34
backward	0.96	0.77	0.86	35
left	0.84	0.76	0.80	34
stop	0.61	0.91	0.73	33
accuracy			0.85	169
macro avg	0.88	0.85	0.85	169
weighted avg	0.88	0.85	0.85	169
Best params: {'alpha': 0.55}				

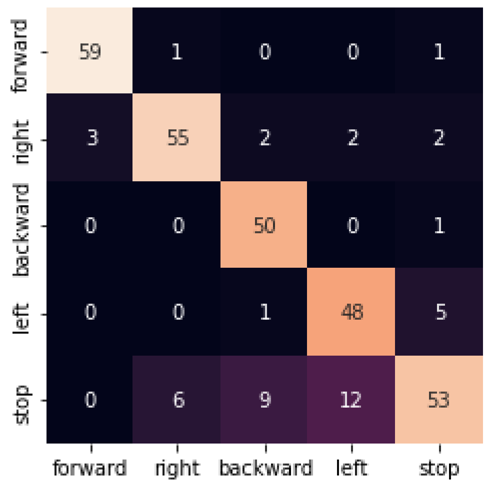
Table 30. MNB-Gestures Test Prediction Result



Figure#38: MNB-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.97	0.95	0.96	62
right	0.86	0.89	0.87	62
backward	0.98	0.81	0.88	62
left	0.89	0.77	0.83	62
stop	0.66	0.85	0.75	62
accuracy			0.85	310
macro avg	0.87	0.85	0.86	310
weighted avg	0.87	0.85	0.86	310
Best params: {'alpha': 0.55}				

Table 31. MNB-Gestures Train Prediction Result

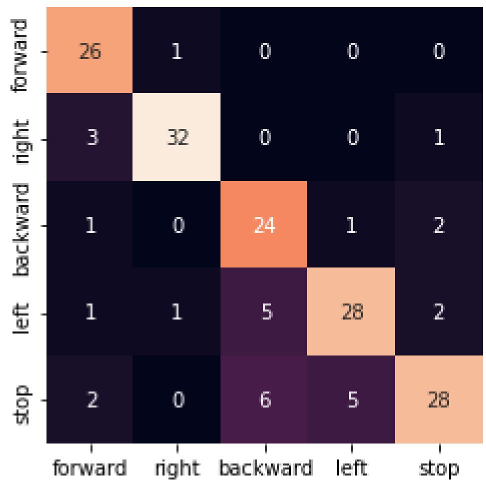


Figure#39: MNB-Gestures Heat Map of Train Prediction

Decision Tree Classification

	precision	recall	f1-score	support
forward	0.96	0.79	0.87	33
right	0.89	0.94	0.91	34
backward	0.86	0.69	0.76	35
left	0.76	0.82	0.79	34
stop	0.68	0.85	0.76	33
accuracy			0.82	169
macro avg	0.83	0.82	0.82	169
weighted avg	0.83	0.82	0.82	169
Best params: {'criterion': 'gini', 'max_depth': 7, 'min_samples_split': 2, 'splitter': 'random'}				

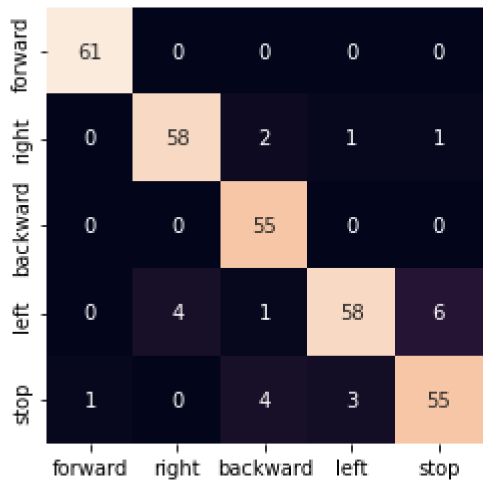
Table 32. DTC-Gestures Test Prediction Result



Figure#40: DTC-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	1.0	0.98	0.99	62
right	0.94	0.94	0.94	62
backward	1.0	0.89	0.94	62
left	0.84	0.94	0.89	62
stop	0.87	0.89	0.88	62
accuracy			0.93	310
macro avg	0.93	0.93	0.93	310
weighted avg	0.93	0.93	0.93	310
Best params: {'criterion': 'gini', 'max_depth': 7, 'min_samples_split': 2, 'splitter': 'random'}				

Table 33. DTC-Gestures Train Prediction Result

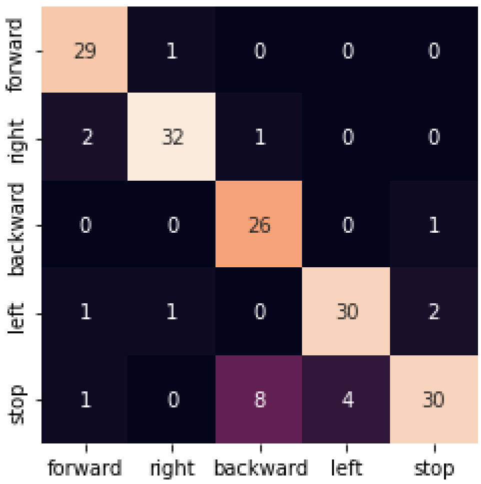


Figure#41: DTC-Gestures Heat Map of Train Prediction

K-Nearest Neighbor Classification

	precision	recall	f1-score	support
forward	0.97	0.88	0.92	33
right	0.91	0.94	0.93	34
backward	0.96	0.74	0.84	35
left	0.88	0.88	0.88	34
stop	0.70	0.91	0.79	33
accuracy			0.87	169
macro avg	0.88	0.87	0.87	169
weighted avg	0.89	0.87	0.87	169
Best parameter result of gridSearch: {'algorithm': 'ball_tree', 'n_neighbors': 4, 'weights': 'uniform'}				

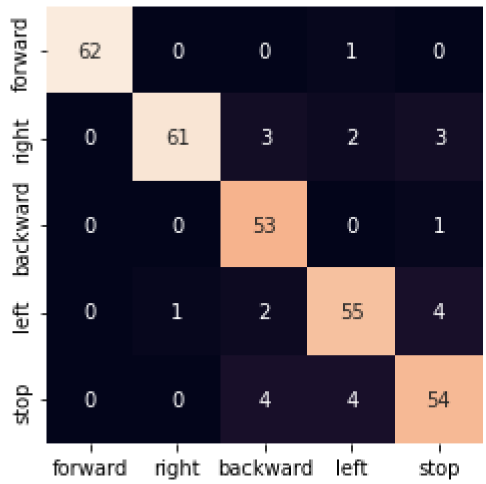
Table 34. KNN-Gestures Test Prediction Result



Figure#42: KNN-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	0.98	1.0	0.99	62
right	0.88	0.98	0.93	62
backward	0.98	0.85	0.91	62
left	0.89	0.89	0.89	62
stop	0.87	0.87	0.87	62
accuracy			0.92	310
macro avg	0.92	0.92	0.92	310
weighted avg	0.92	0.92	0.92	310
Best parameter result of gridSearch: {'algorithm': 'ball_tree', 'n_neighbors': 4, 'weights': 'uniform'}				

Table 35. KNN-Gestures Train Prediction Result

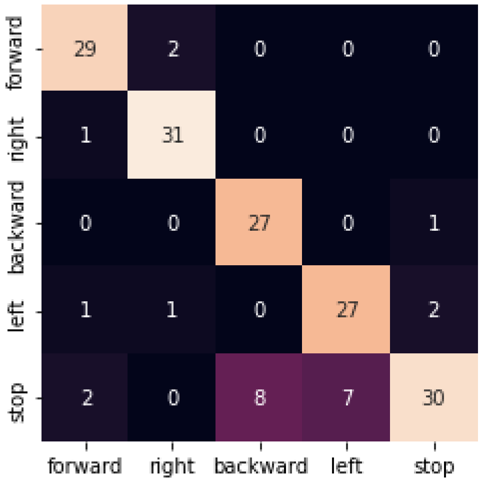


Figure#43: KNN-Gestures Heat Map of Train Prediction

Voting Classification

	precision	**recall**	**f1-score**	**support**
forward	0.94	0.88	0.91	33
right	0.97	0.91	0.94	34
backward	0.96	0.77	0.86	35
left	0.87	0.79	0.83	34
stop	0.64	0.91	0.75	33
accuracy			0.85	169
macro avg	0.88	0.85	0.86	169
weighted avg	0.88	0.85	0.86	169
Used Methods: SVM, Logistic regression, gradient boosting classification				

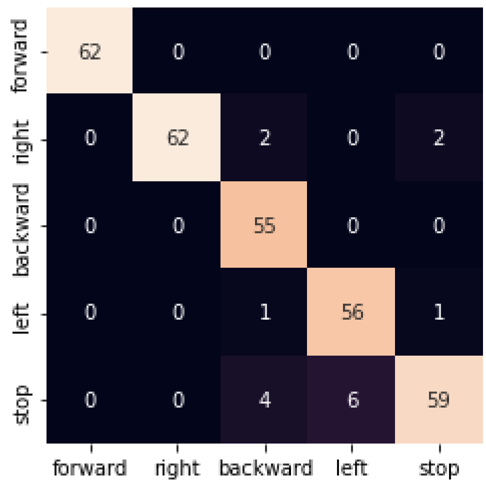
Table 36. VC-Gestures Test Prediction Result



Figure#44: VC-Gestures Heat Map of Test Prediction

	precision	recall	f1-score	support
forward	1.0	1.0	1.0	62
right	0.94	1.0	0.97	62
backward	1.0	0.89	0.94	62
left	0.97	0.90	0.93	62
stop	0.86	0.95	0.90	62
accuracy			0.95	310
macro avg	0.95	0.95	0.95	310
weighted avg	0.95	0.95	0.95	310
Used Methods: SVM, Logistic regression, gradient boosting classification				

Table 37. VC-Gestures Train Prediction Result



Figure#45: VC-Gestures Heat Map of Train Prediction

7. CONCLUSION

The aim of this project was to help people with physical problems to live more comfortably and to produce solutions to their problems, even if to some extent. In addition, it can be used on any person and it is possible to give people abilities as if they have an extra arm and leg. But it should not be denied that there is a learning process for this. It can be thought of as driving a car. We do not know how to drive from birth, but with practice and time we can achieve this. The same is true for such systems.

From a technical point of view, it would be a more logical and more realistic approach to go with EMG-oriented devices with such cheap devices. 40% accuracy obtained with EEG is not at a level that can compete with 80-90% accuracy obtained with EMG. Therefore, it does not make much sense to set up a comprehensive EEG system with the inexpensive devices of OpenBCI, but it is also a difficult task to perform.

7.1. Life-Long Learning

During the development of this project, the students, who had the opportunity to work on more than one technology, were able to obtain very good information and experiences. The use of hardware such as RaspberryPi board, OpenBCI EEG Headset and PiCar contributed greatly to both the students and the project. Students used a lot of Python libraries and thus they had opportunities to improve themselves in this field. Only 2 different large libraries were used for the EEG Headset (Brainflow and MNE). In the future, it is possible to say that they have general knowledge if they work on the processing of brain signals. They learned how a drone car that can be controlled with RaspberryPi can be controlled with another device together with a connection, and they succeeded in moving objects with the help of this method. Thus, they were able to establish a connection between two different hardware and observe it. They also learned the collection and analysis of EEG data in this process. They ensured that these data were collected meaningfully and modeled using ML algorithms, making these models usable. Thus, they had the opportunity to work in more than one field and learned to make connections between them. The research carried out during these studies allowed them to have experience about the tricks of doing scientific research. Thus, they have improved in reaching the right resources and integrating them correctly into the systems.

7.1.1 Future Work

First of all, let's consider the developmental aspects of the MI (Motor/Mental Imagery) part. Some of the things that can be done to improve this part of the project are using 200ms chunks instead of currently using 500ms chunks so this system can produce better results with less time spent collecting data. Thus, it is possible to collect more data in frequency ranges such as alpha and beta. But in order to achieve this, the system with a 125 Hz sampling rate and 16 channels must be increased to at least 250Hz sampling rate. Because in order for the obtained data to go through certain processes, it is necessary to collect more than a certain number of data. For example, if we divide 125Hz into 2 different parts as 500ms, one of the flat data we will have will be around 62-63. While functions such as denoising can be applied to this amount of data, it is not possible to apply this function to data collected under a certain number, for example around 30-40, with the Brainflow library. In other words, we cannot divide this system into 200ms parts due to hardware inadequacy. In addition, the use of professional devices that provide more channels and more sampling rates will provide great convenience during the development of this project. The fact that this equipment is dry-headed causes it to be exposed to more noise. Although it is a system that can be supported with gel, it is very possible to understand that the subjects did not prefer it in terms of cleanliness.

How is it possible to make improvements on the models of the system developed using gestures? First of all, different gestures should be tried and the situations where the distinction between directions can be made easiest should be observed. It should be noted that this can vary from person to person. It is also necessary to say that in projects like this, situations such as the shape of the head of the person and the inability to put the device back, in the same way, change the way the system works, either positively or negatively. For this, using the gyroscope features of the device and with it, whether the device is installed correctly or not can be tested. Regarding the data collection process, in order to increase the success achieved with a 5-minute data collection session, needs such as more sampling rates and more channels in the hardware area remain valid for this system as well. It is of great importance to keep these data collection periods short and they should not be extended. Because the people who will use these systems are dealing with certain diseases, it is unrealistic to collect data for a long time both for them and for the operators who will help them during the development of this system. We recommend that students or academics who want to continue this project in the future think about them.

In addition to these, the collection of more data and the creation of a pool dataset are among the future works of this project. Collecting large numbers of subjects and data for both systems can increase the validity and success rates of the system. As a disadvantage of the time we spent on this system to collect data correctly, we did not have a chance to collect much data. In addition, the analysis of the advantages and disadvantages of constantly collecting data in different ways put us in a difficult position on this path.

7.2. Professional and Ethical Responsibilities of Engineers

It is possible to say that this project has many areas suitable for ethical evaluation. First of all, it is important that the data of the people who will use this system are protected safely and that it is not shared with other people unless they want to. In order to ensure this condition, care is taken to keep the information locally, so that the data is not available on the internet. The data that people collect to develop models is safely stored on their computers.

In addition to the security of data, since such applications must always be open on computers, there must be an application that must be open all the time in order to take an instant action, and there must be no malicious software in this application. For this, the coding must be changed by reliable and audited people.

When using this system, it is strongly requested not to take any risky actions. To give an example, it may be risky to connect this system to a wheelchair to get around on the street, since this system does not have a success rate close to 100%. In such cases, a helper should always be present with the wheelchair user. We recommend that this system be used in a simpler way, such as moving the cursor on the screen, but in a way that can touch people's lives.

7.3. Contemporary Issues

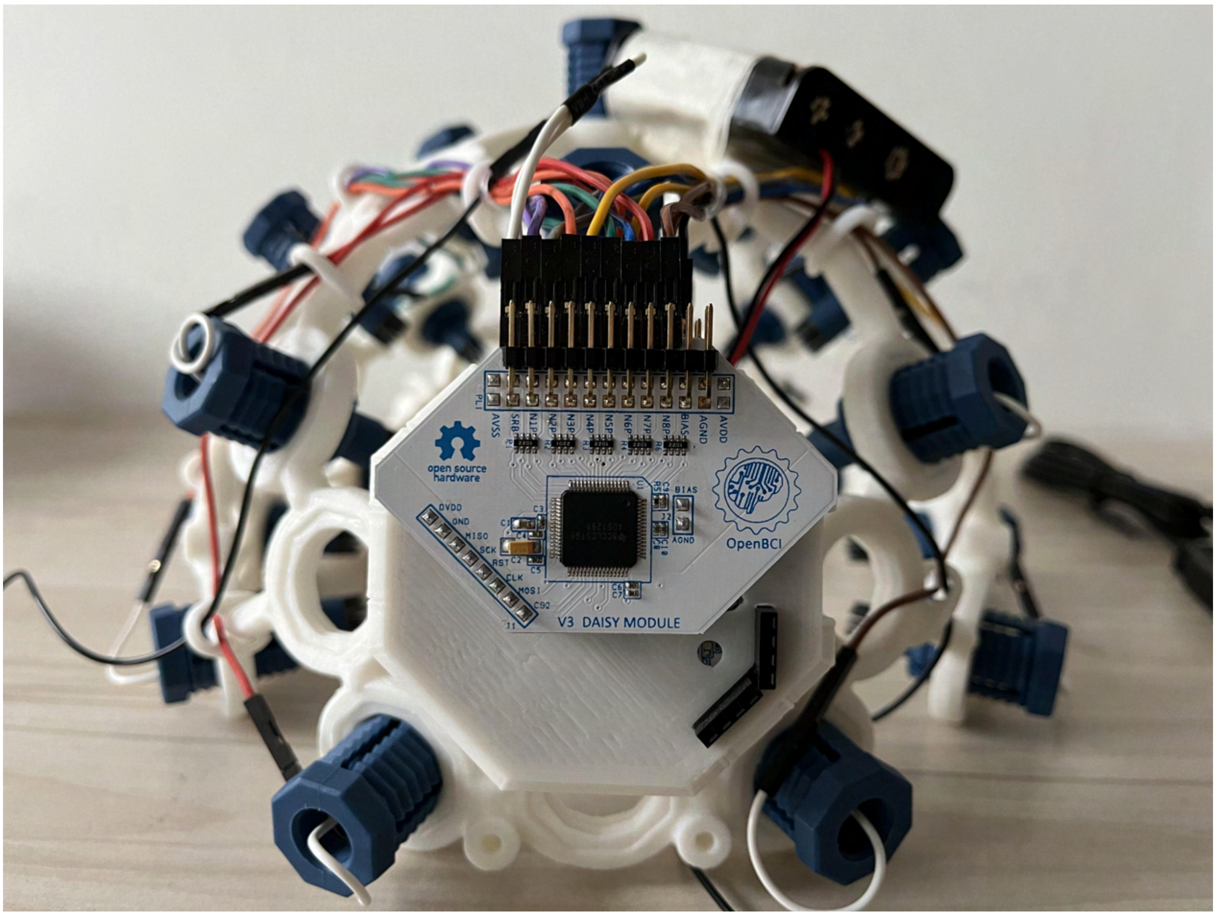
It is possible to say that this project currently has a lot of problems. Because if we consider that even the low-quality headsets of OpenBCI are very expensive and we add that it is certain that not everyone can afford them, we are faced with big problems. In addition to these, different engines and systems will be needed for hardware mobilization in such projects, so the costs of these will be added to the system itself. It is also necessary to maintain them regularly and ensure that the devices remain in continuous operation. It is possible to foresee that the labor force and money spent for these will be too much for now. But in the future, it is possible to expect such problems to decrease with the cheapening of EEG headsets. As such systems become widespread, it will be possible to obtain such systems more cheaply and they will be used more widely.

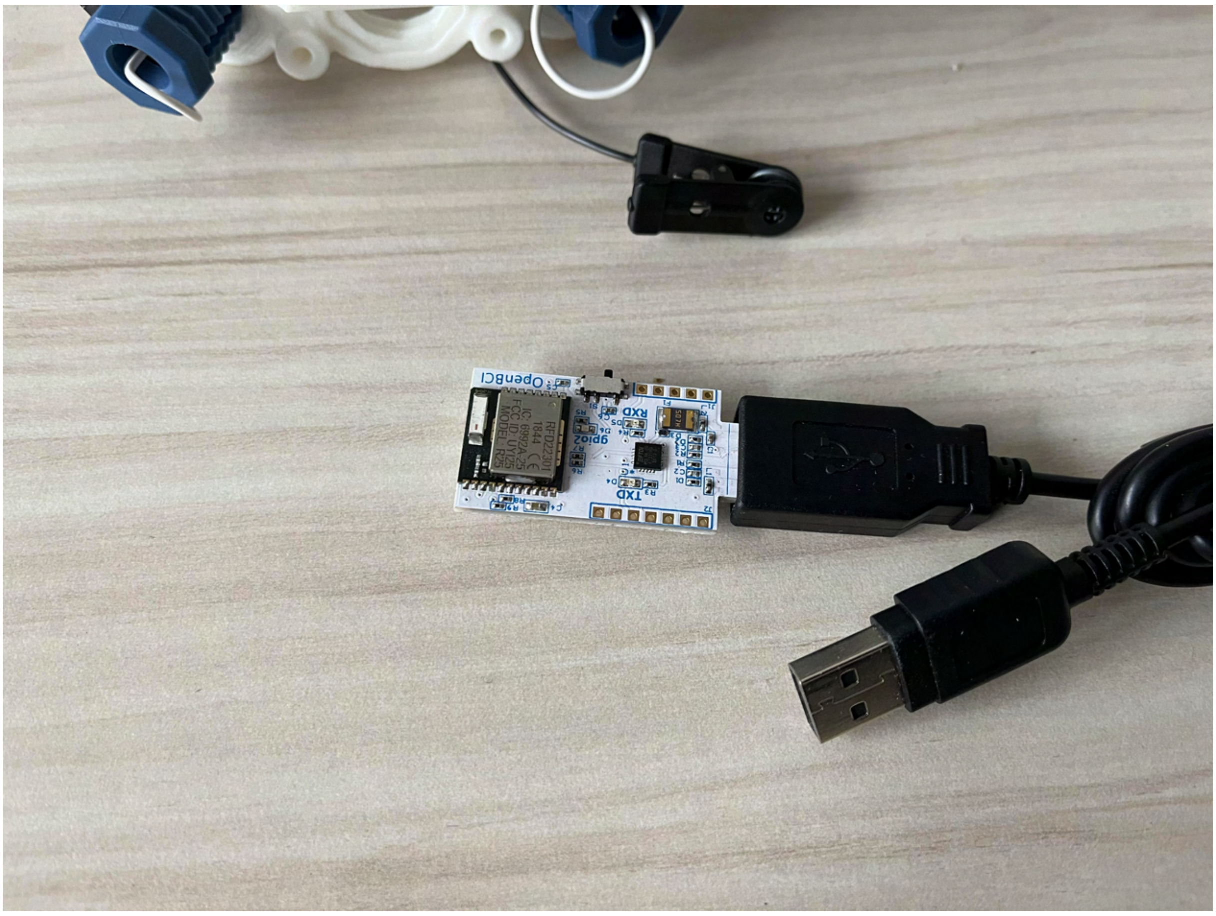
So, is it possible to call it a good system for those who do not have financial problems with such things? This too remains open to debate. Because if such systems are developed using the best EEG headsets, it is almost certain that much better results will be obtained, but while these devices are constantly supervised by someone and used especially on disabled individuals, people who wear these devices for disabled individuals should also know how to do it. It is certain that systems are unlikely to function properly if this is done incorrectly.

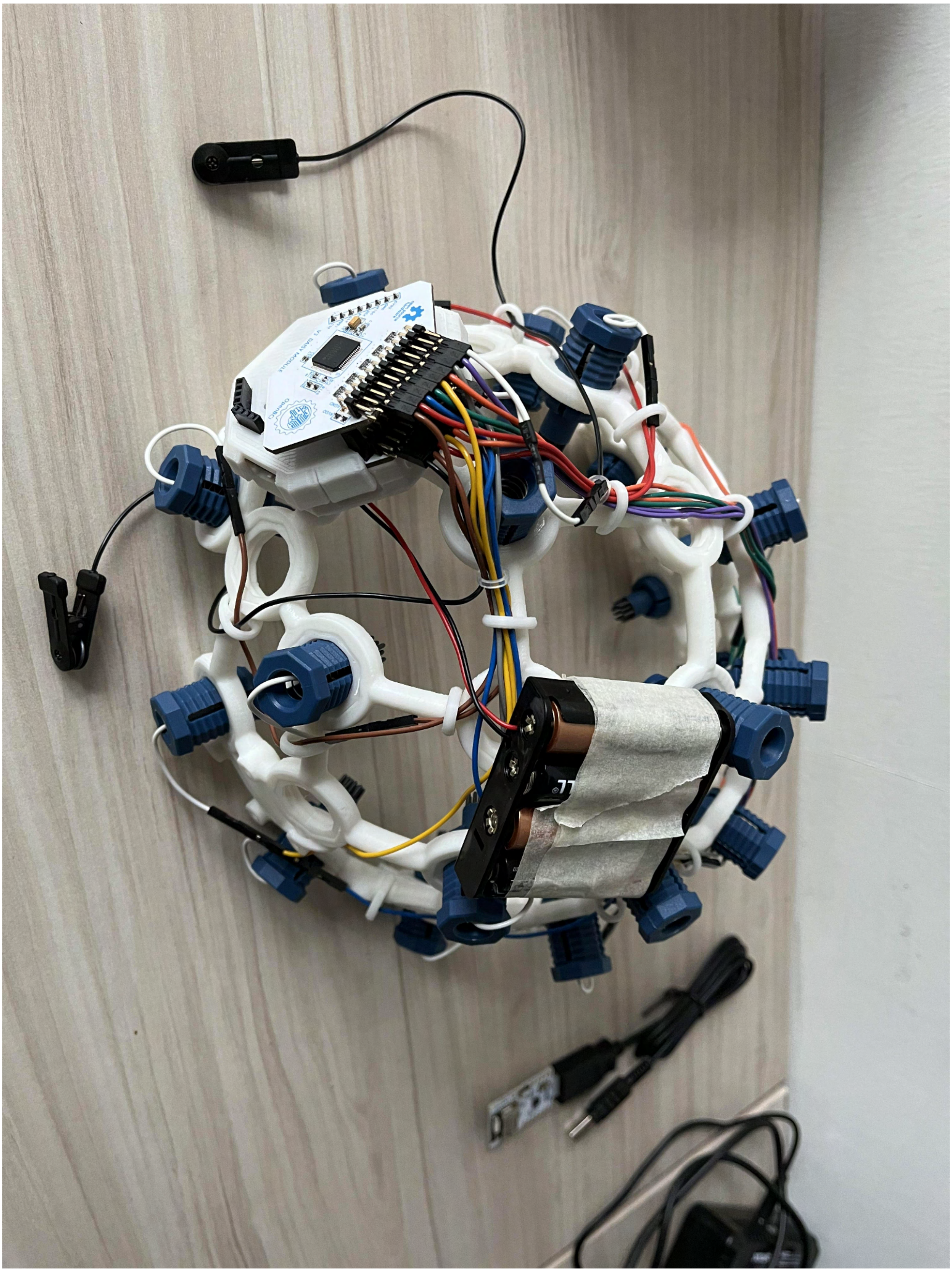
7.4. Team Work

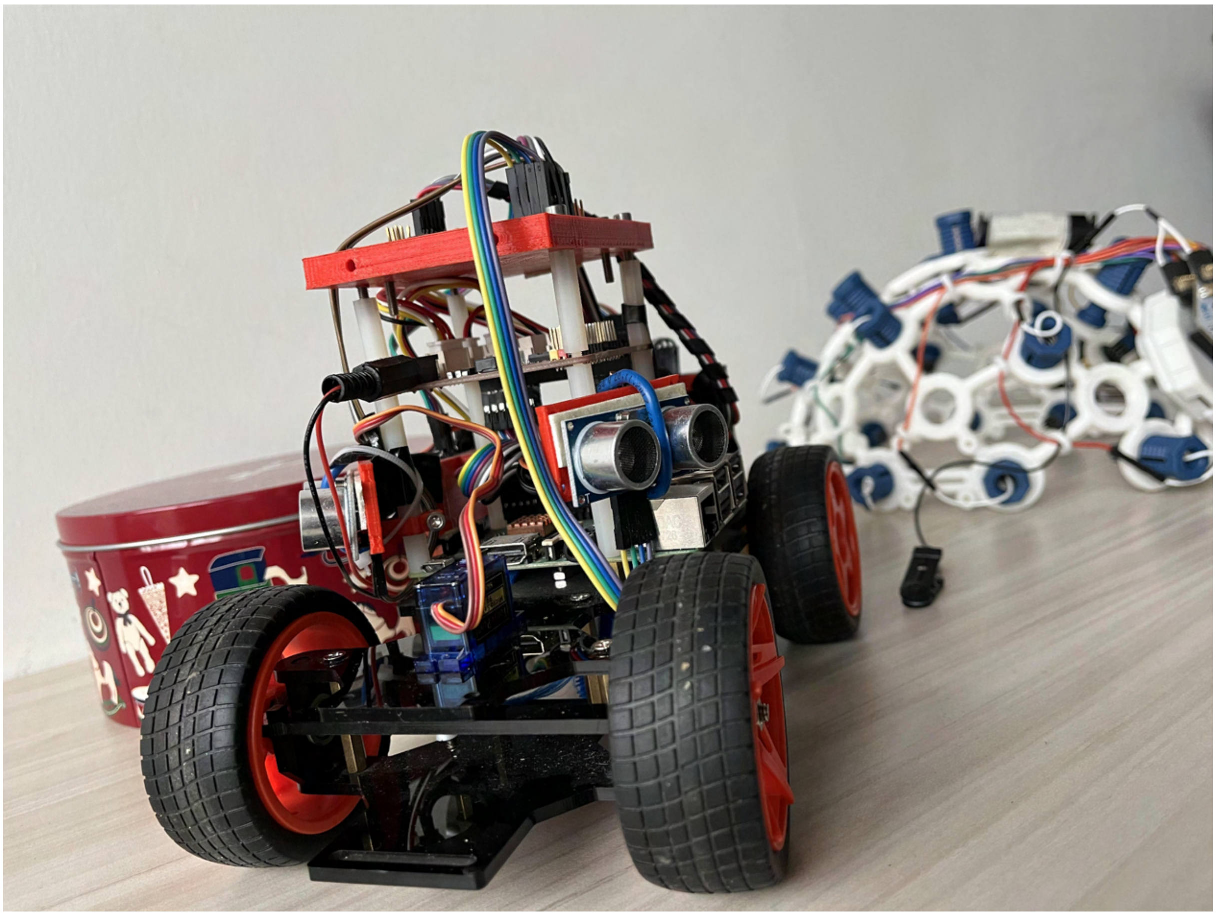
As those working on the project, we were all students. It would not be wrong to say that we worked together at every step of the project. We needed to understand step by step how hardware works. In the part of establishing the connection between the hardware, it was imperative that we all have similar knowledge levels. From the moment we realized this, we decided to move forward together. We also thought together about the solutions to the bugs that needed to be solved. We held physical and online meetings regularly. In general, we can say that we have progressed efficiently.

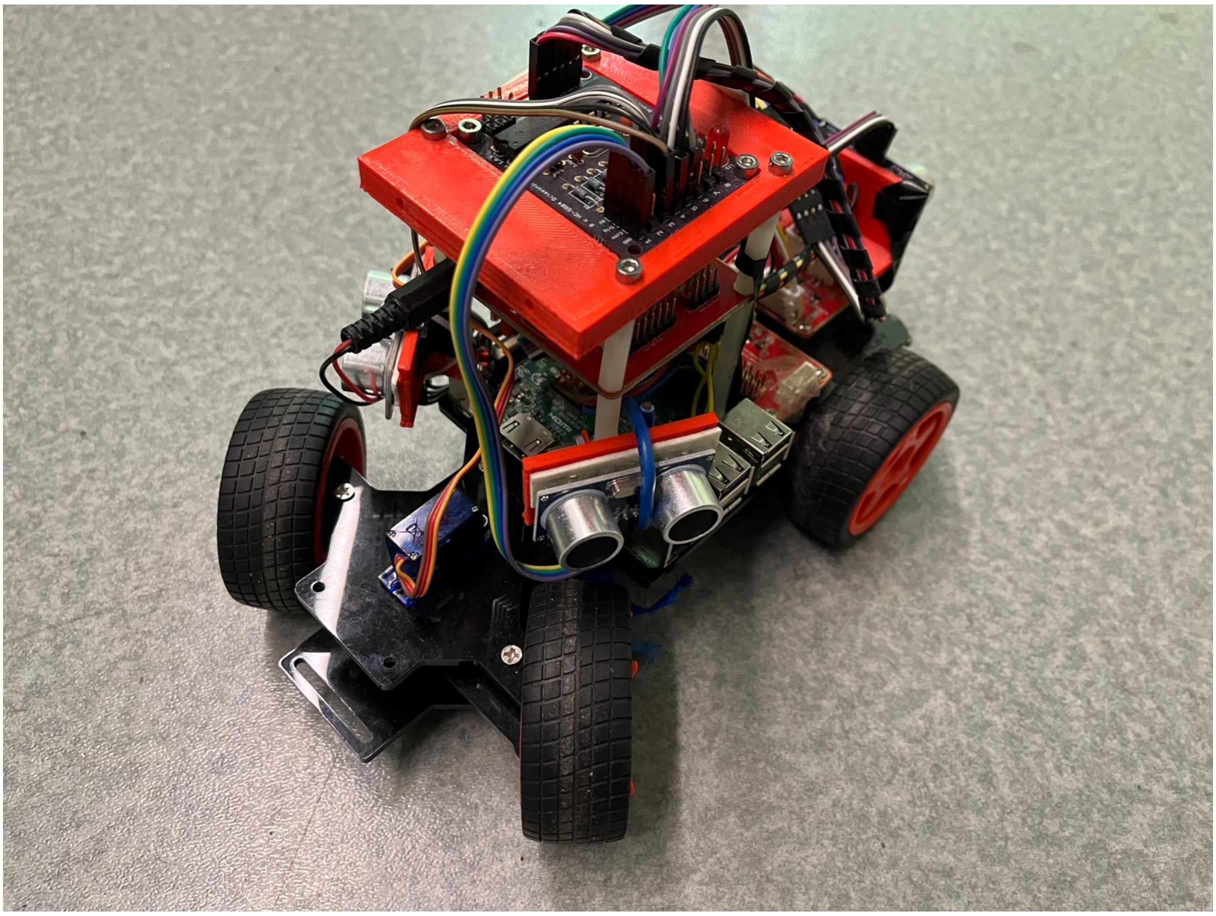
APPENDIX A

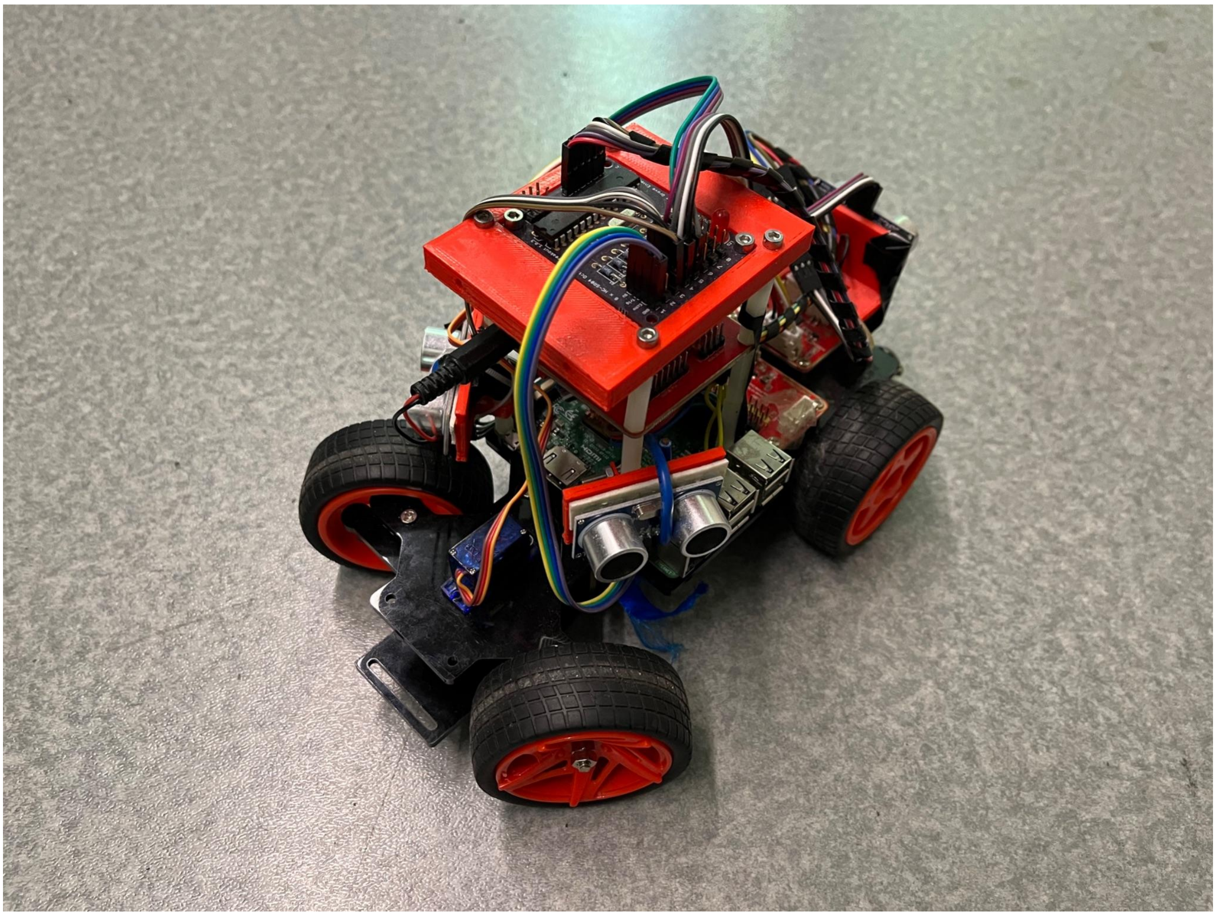

open source
hardware
OpenBCI
V3 DAISY MODULE

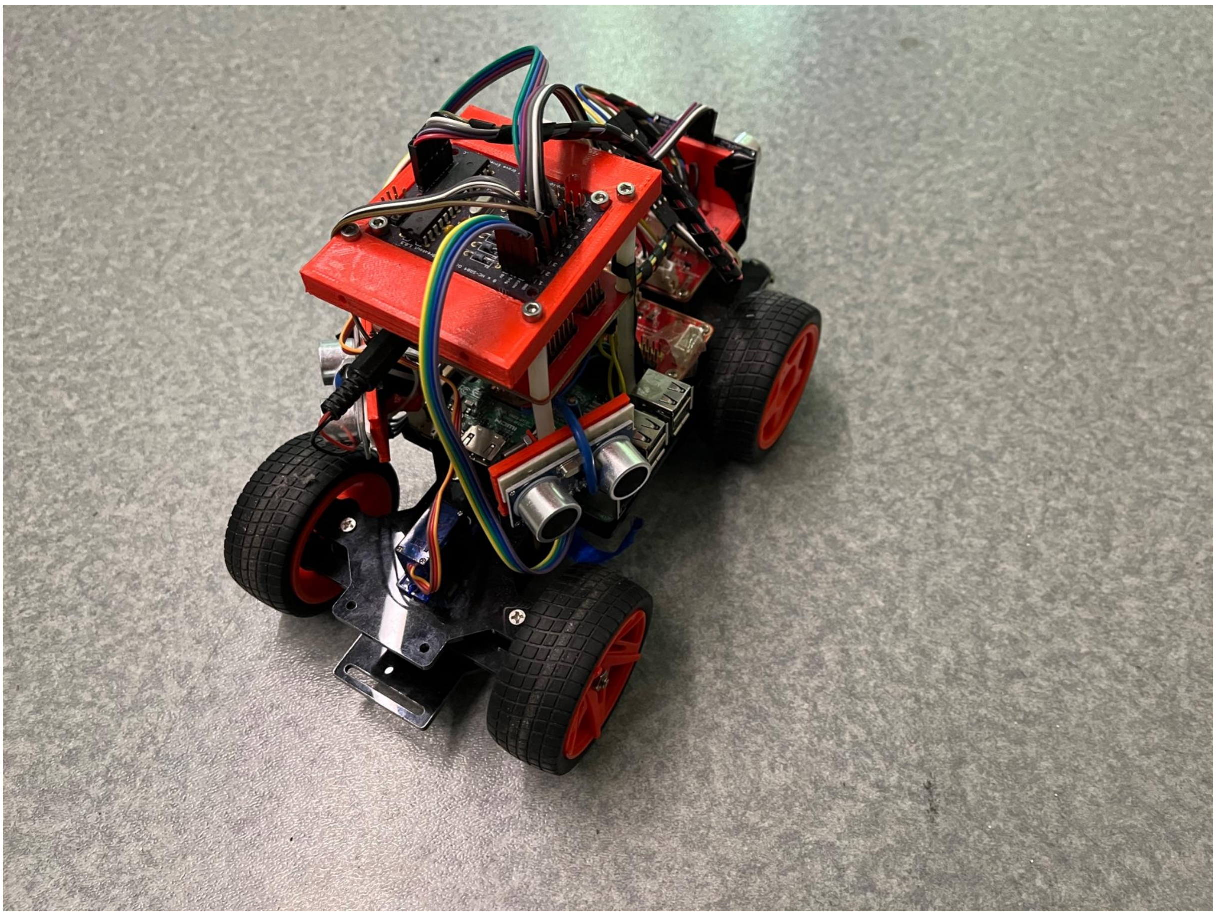

OpenBCI
RXD
gpio2
TXD





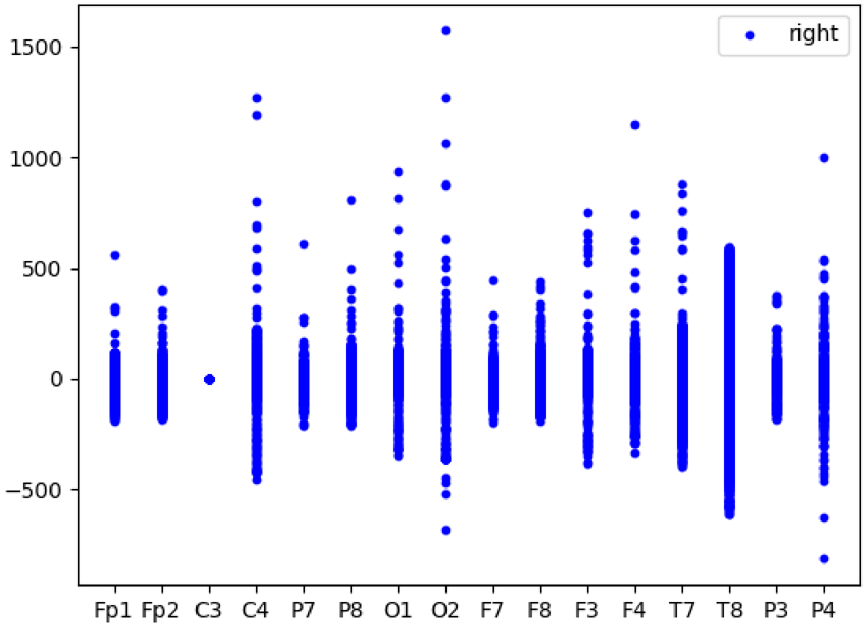


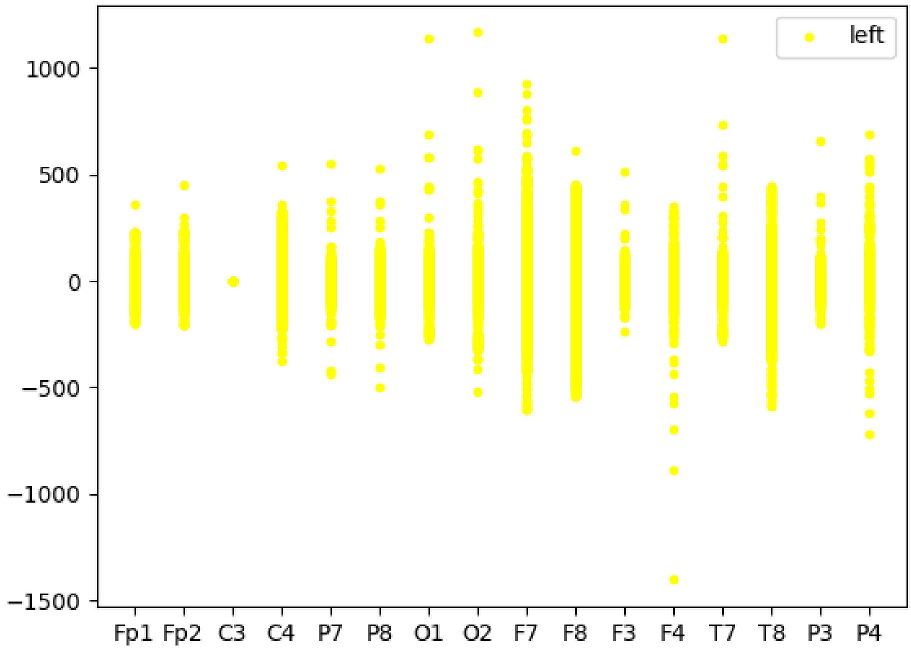


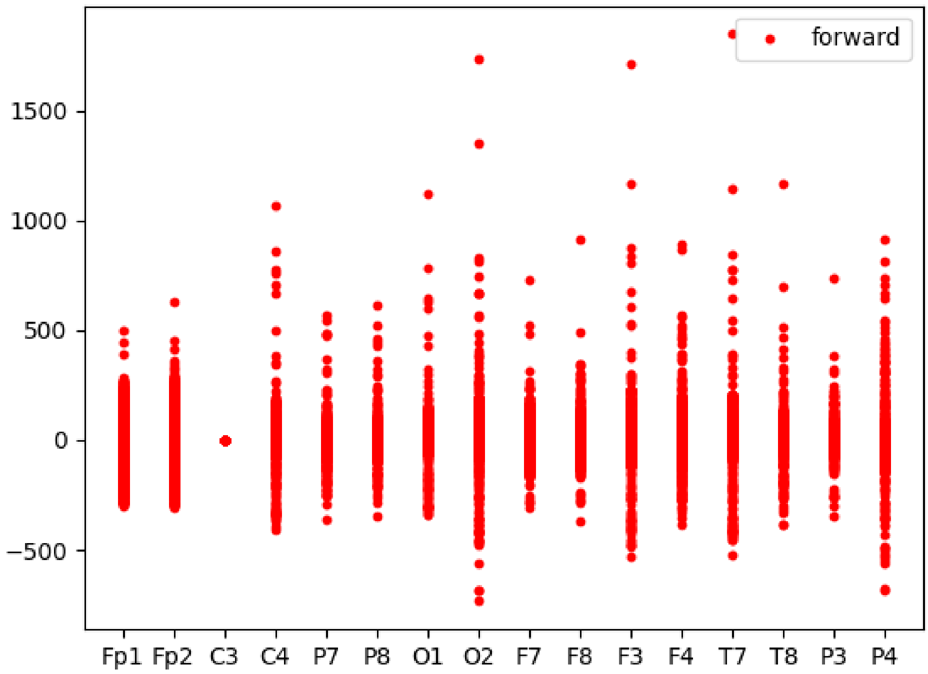


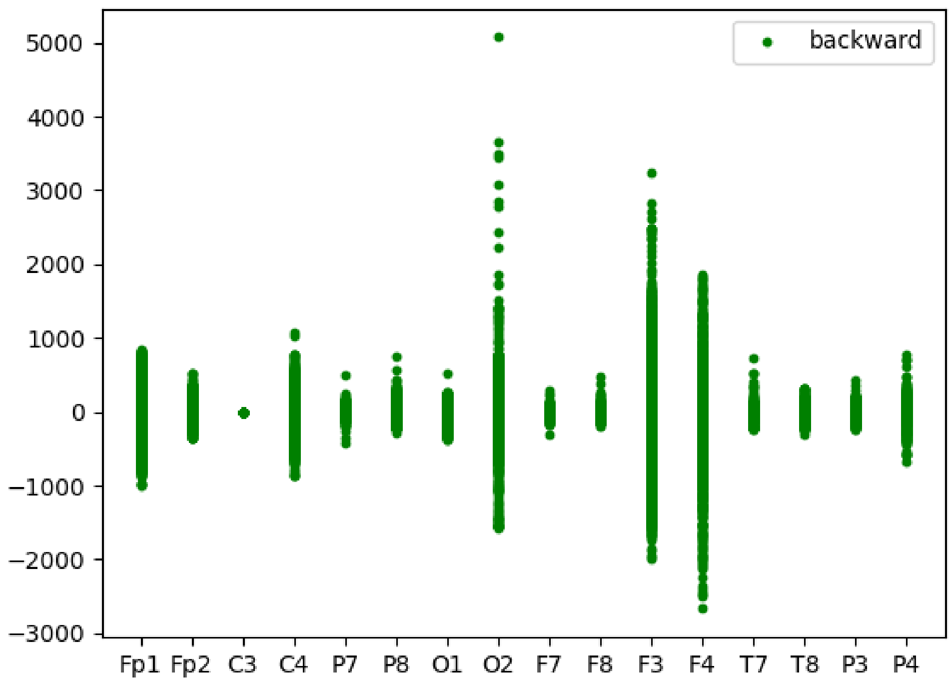
APPENDIX B

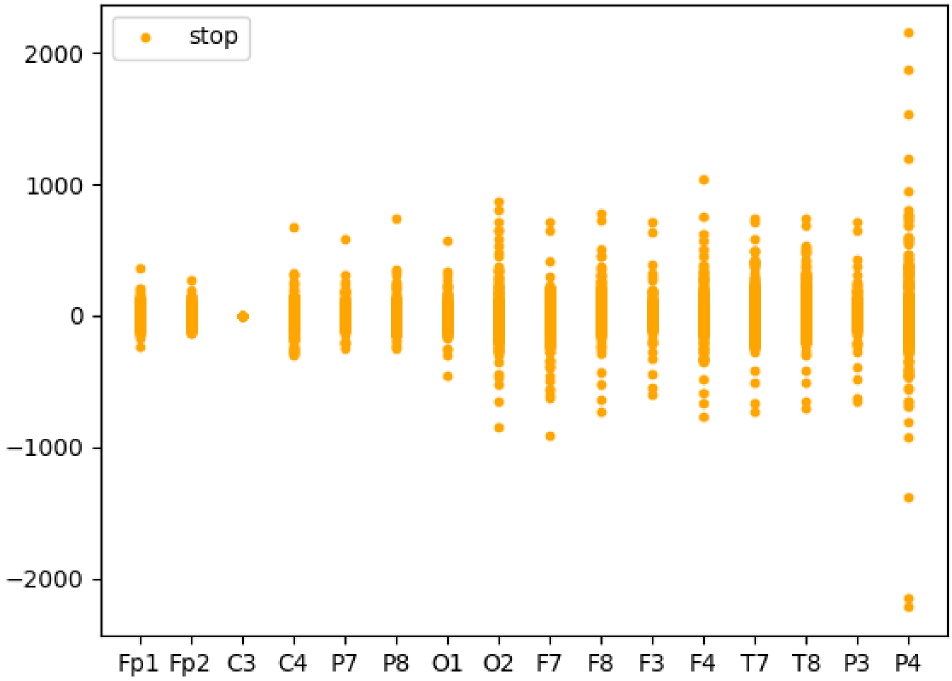
Gesture/EMG Data Visualization

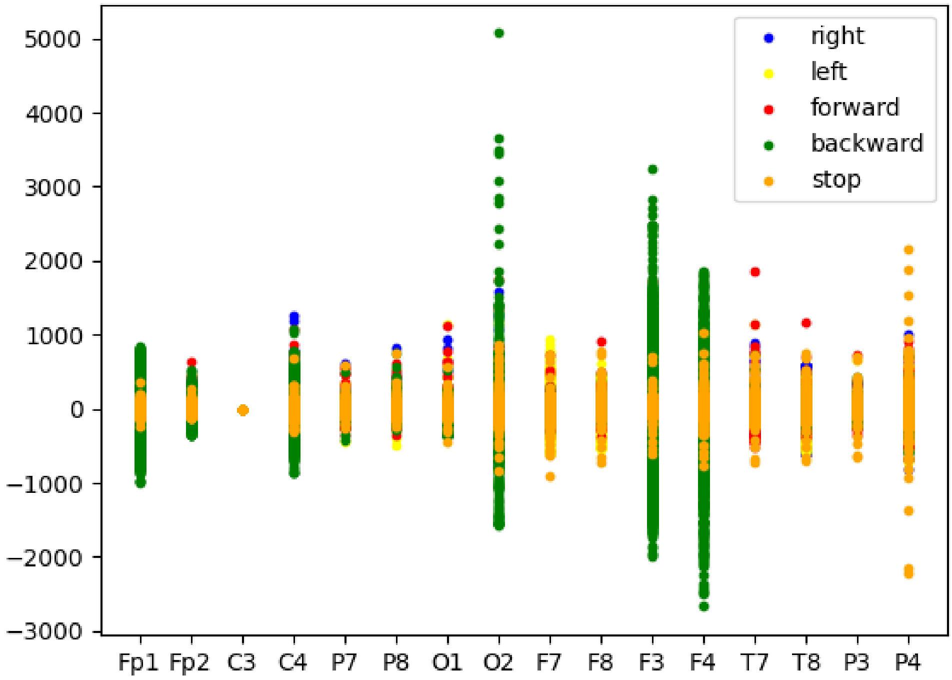


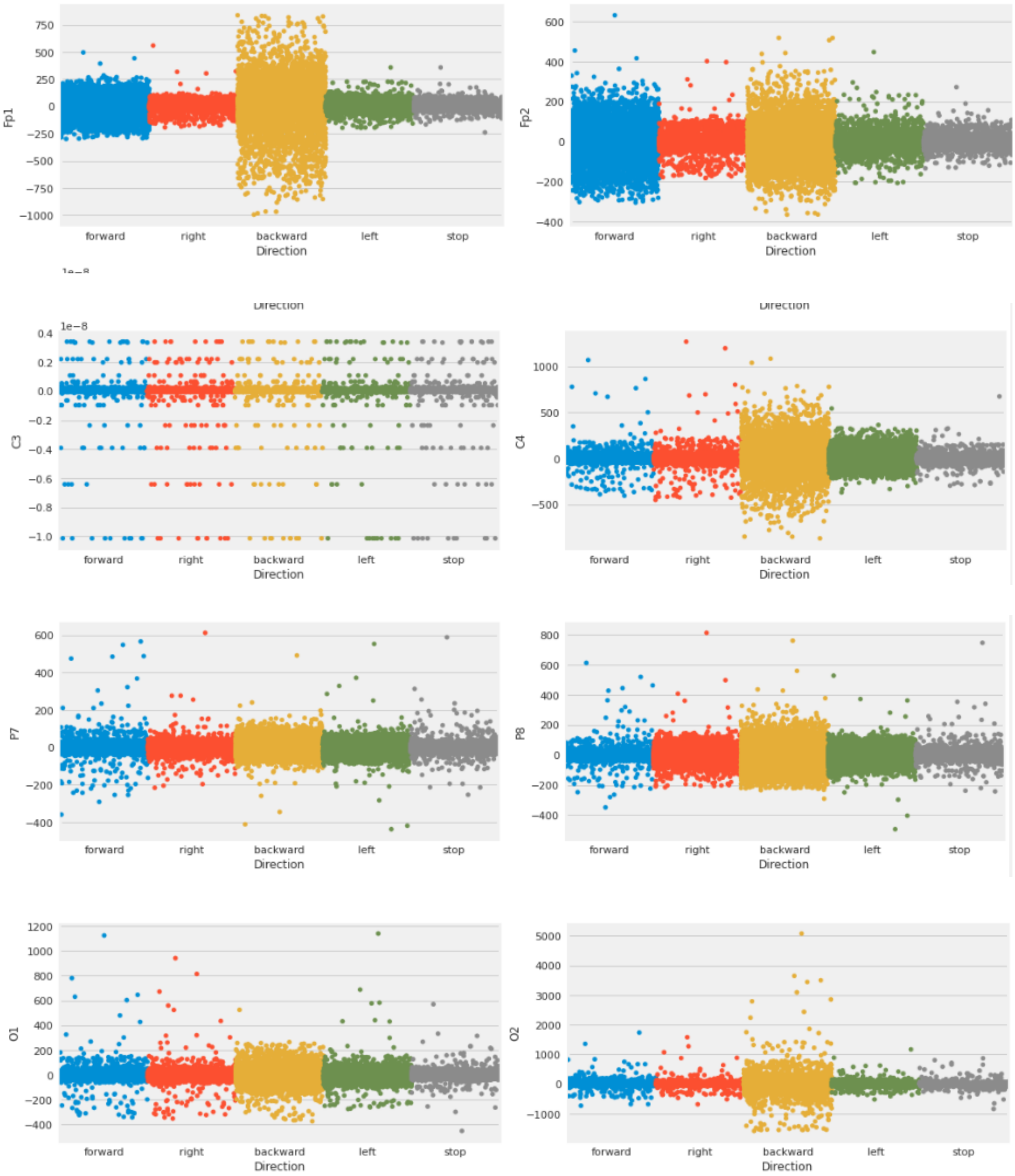



forward
1500
1000
500
0
−500
Fp1 Fp2 C3 C4 P7 P8 O1 O2 F7 F8 F3 F4 T7 T8 P3 P4


backward
5000
4000
3000
2000
1000
0
−1000
−2000
−3000
Fp1 Fp2 C3 C4 P7 P8 O1 O2 F7 F8 F3 F4 T7 T8 P3 P4

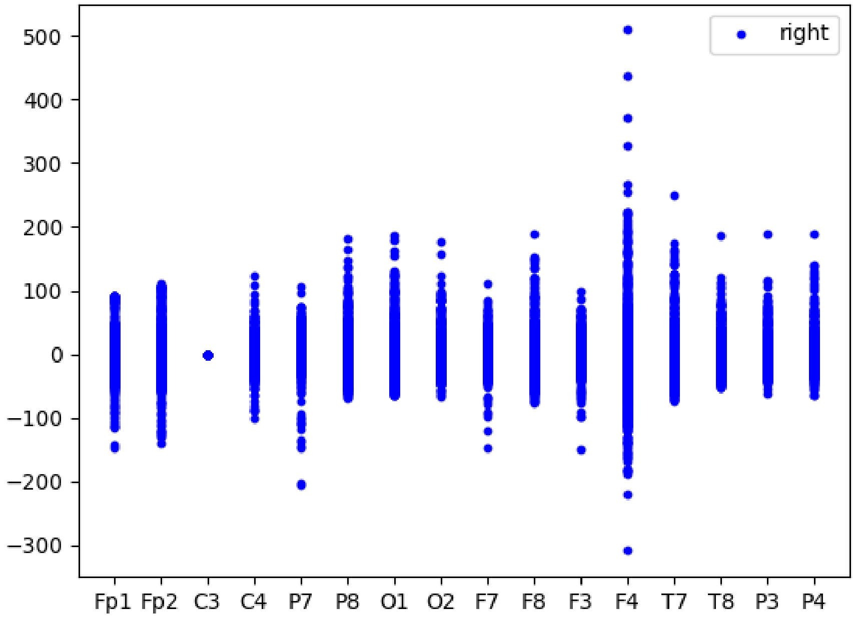

stop
2000
1000
0
−1000
−2000
Fp1 Fp2 C3 C4 P7 P8 O1 O2 F7 F8 F3 F4 T7 T8 P3 P4

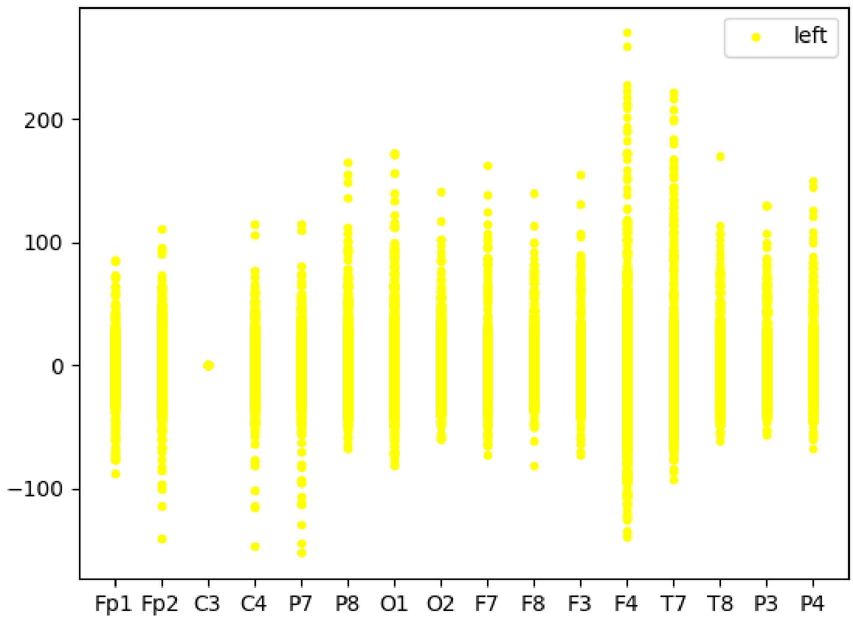

right
left
forward
backward
stop
5000
4000
3000
2000
1000
0
−1000
−2000
−3000
Fp1 Fp2 C3 C4 P7 P8 O1 O2 F7 F8 F3 F4 T7 T8 P3 P4

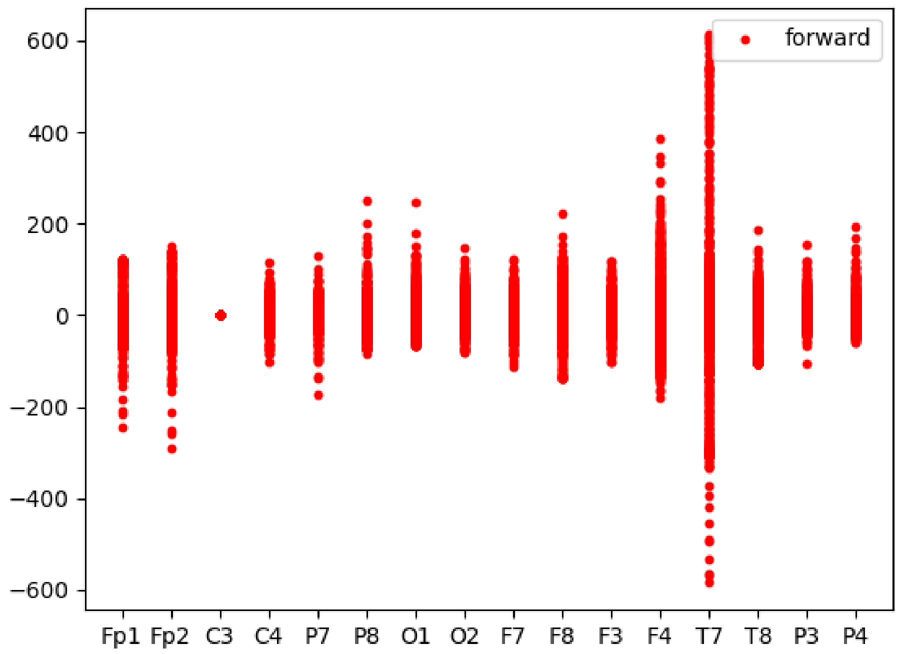

Fp1
Fp2
C3
1e−8
C4
P7
P8
O1
O2
forward
right
backward
left
stop
Direction

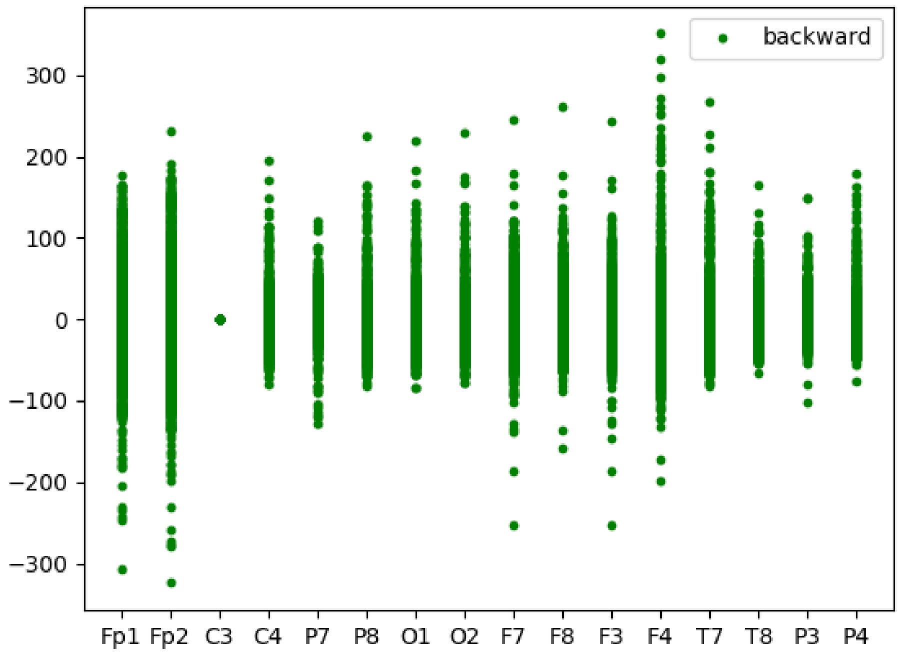

F7
F8
F3
F4
T7
T8
P3
P4
forward
right
backward
left
stop
Direction

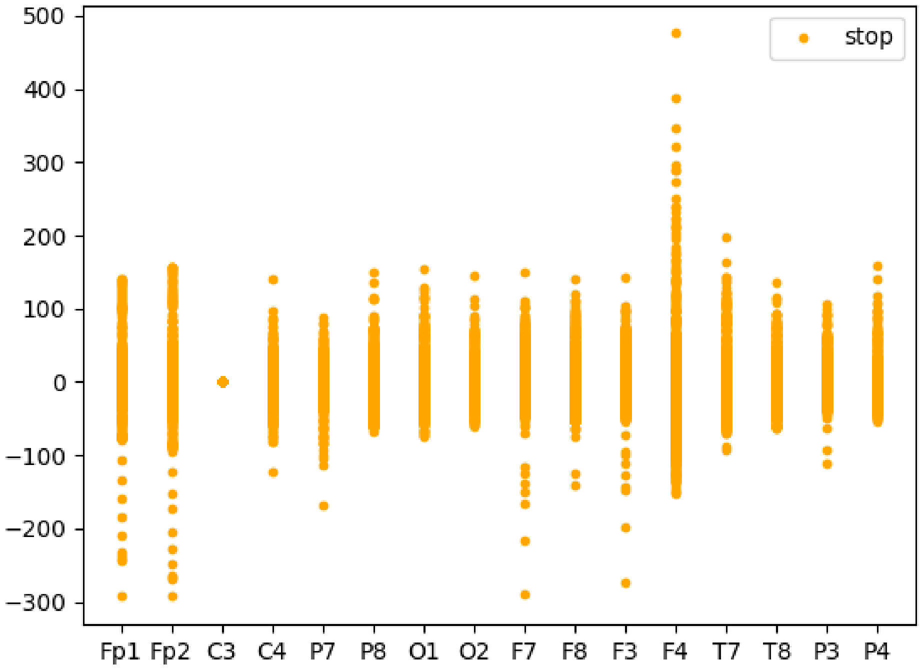
Mental Imagery/EEG Data Visualization

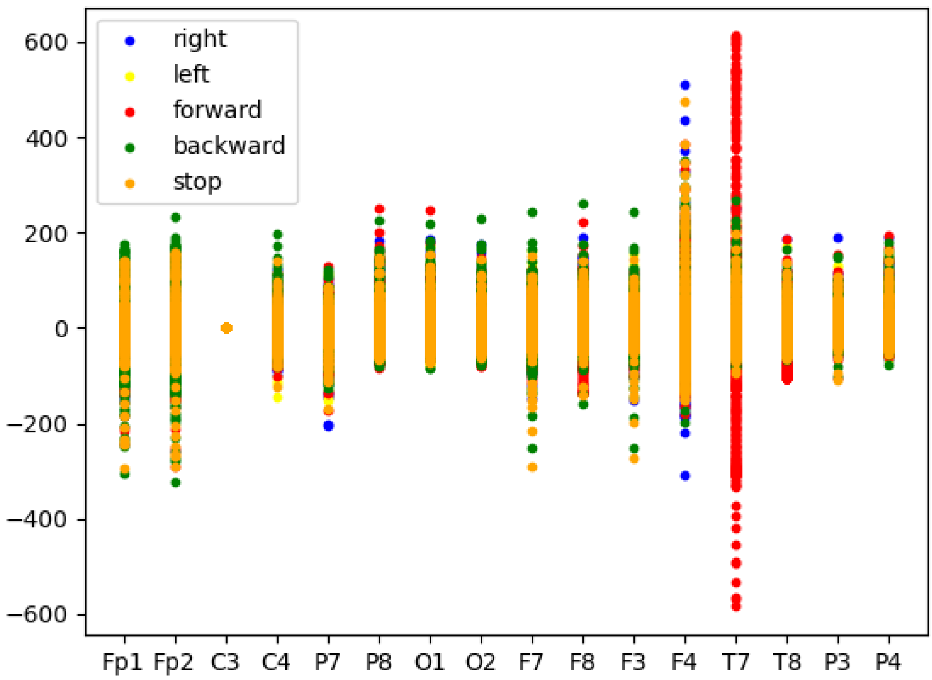


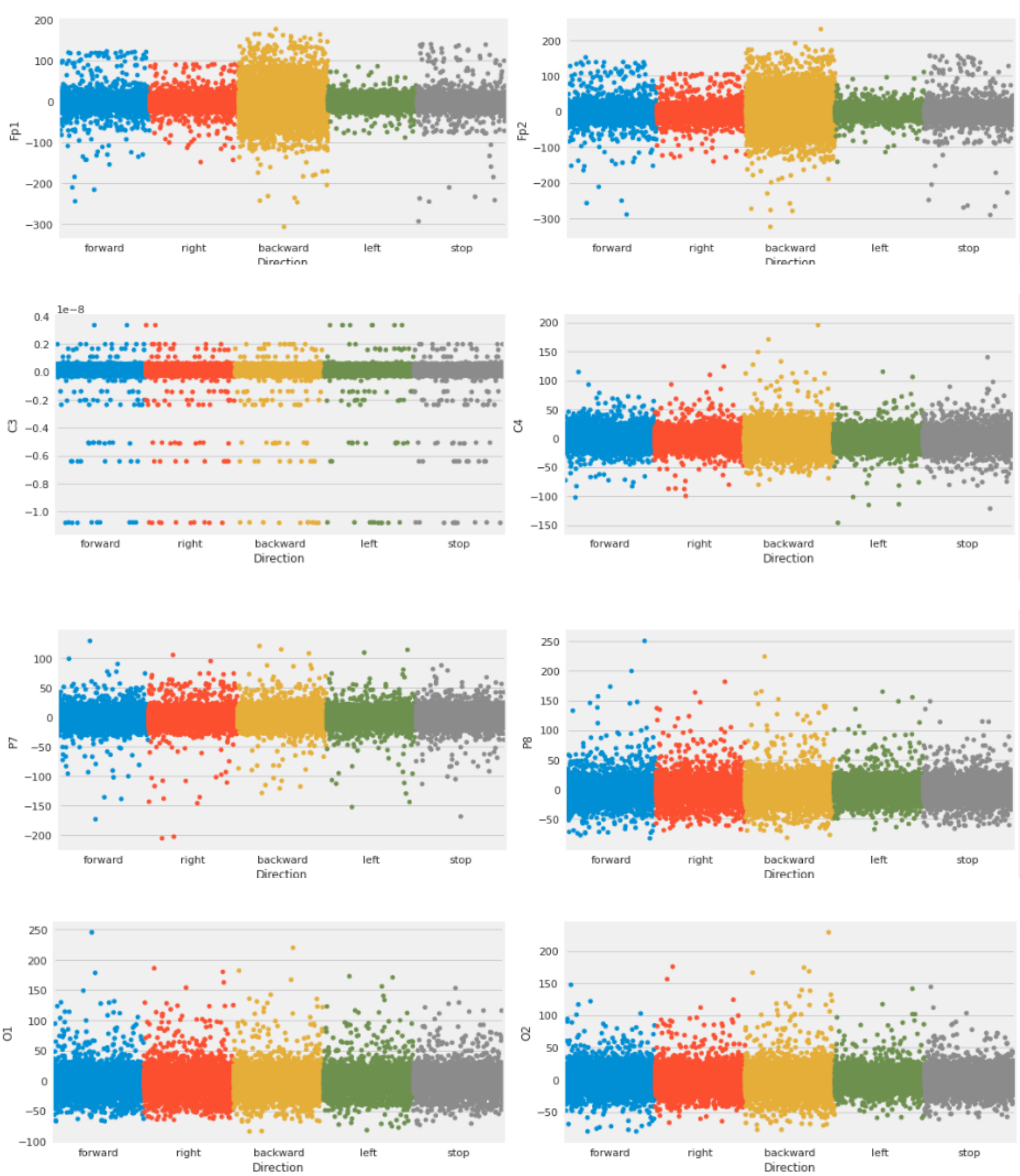



forward
600
400
200
0
−200
−400
−600
Fp1 Fp2 C3 C4 P7 P8 O1 O2 F7 F8 F3 F4 T7 T8 P3 P4

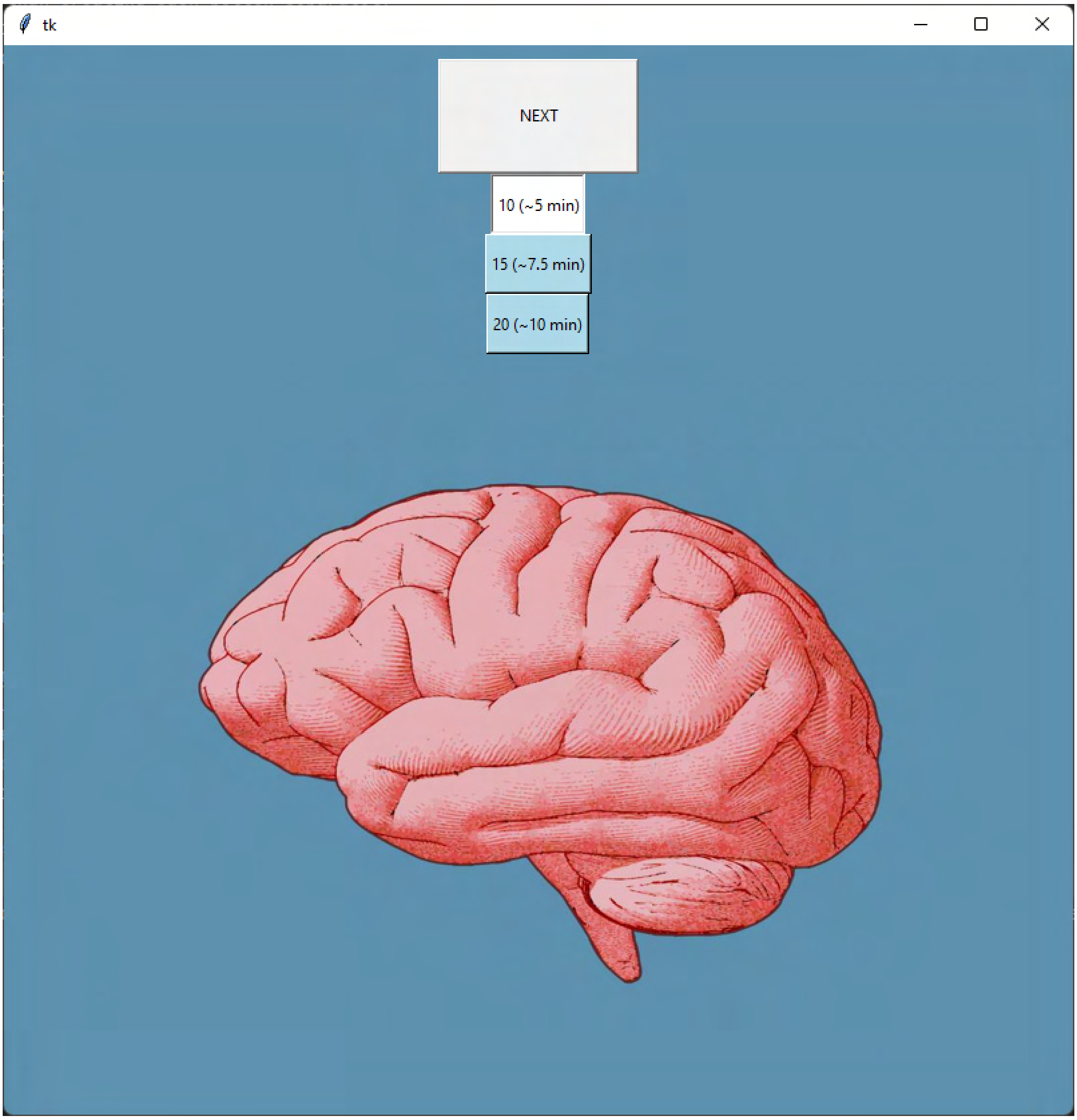

backward
300
200
100
0
−100
−200
−300
Fp1 Fp2 C3 C4 P7 P8 O1 O2 F7 F8 F3 F4 T7 T8 P3 P4

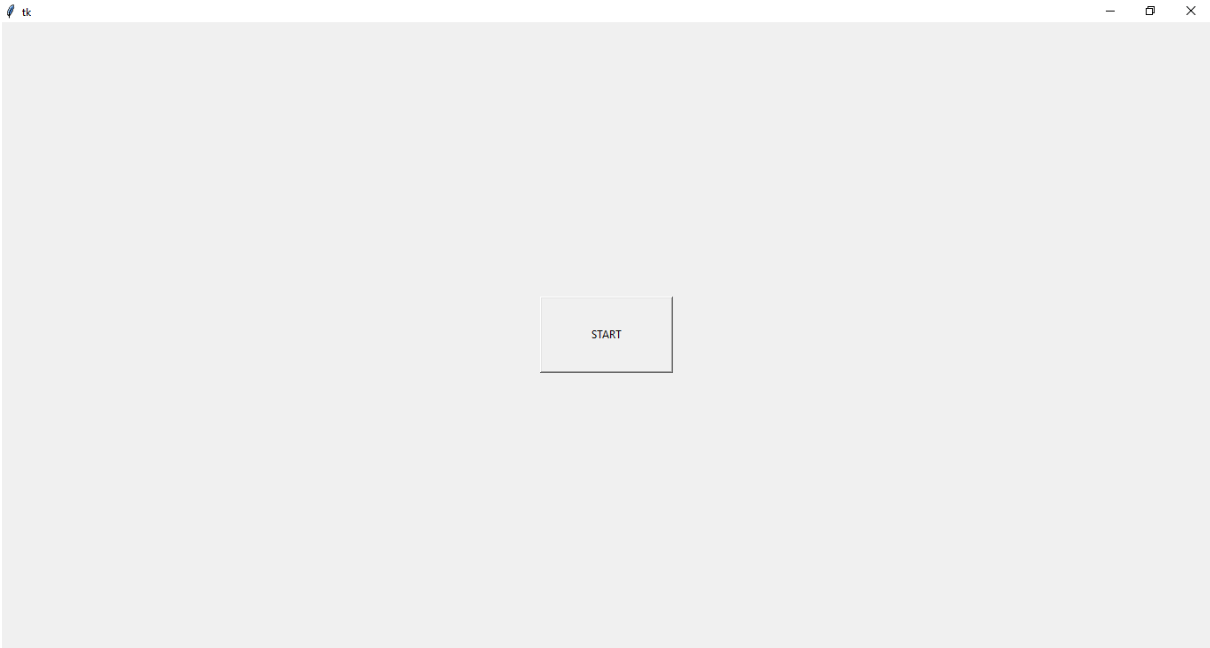

500
400
300
200
100
0
−100
−200
−300
stop
Fp1
Fp2
C3
C4
P7
P8
O1
O2
F7
F8
F3
F4
T7
T8
P3
P4

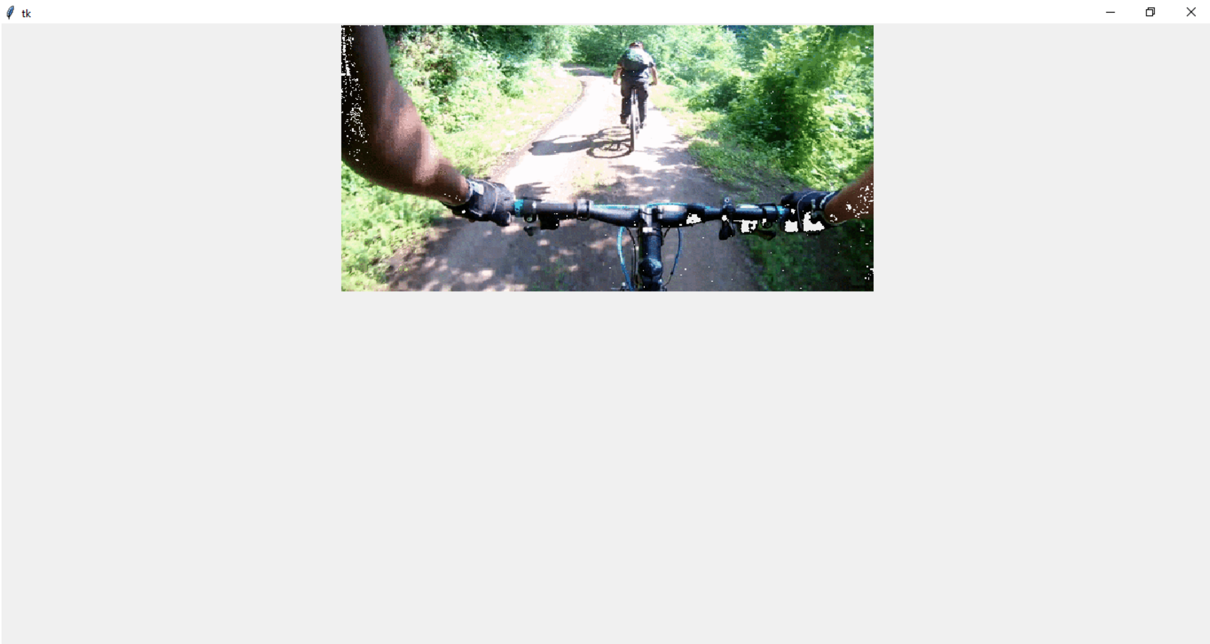

600
400
200
0
−200
−400
−600
right
left
forward
backward
stop
Fp1
Fp2
C3
C4
P7
P8
O1
O2
F7
F8
F3
F4
T7
T8
P3
P4

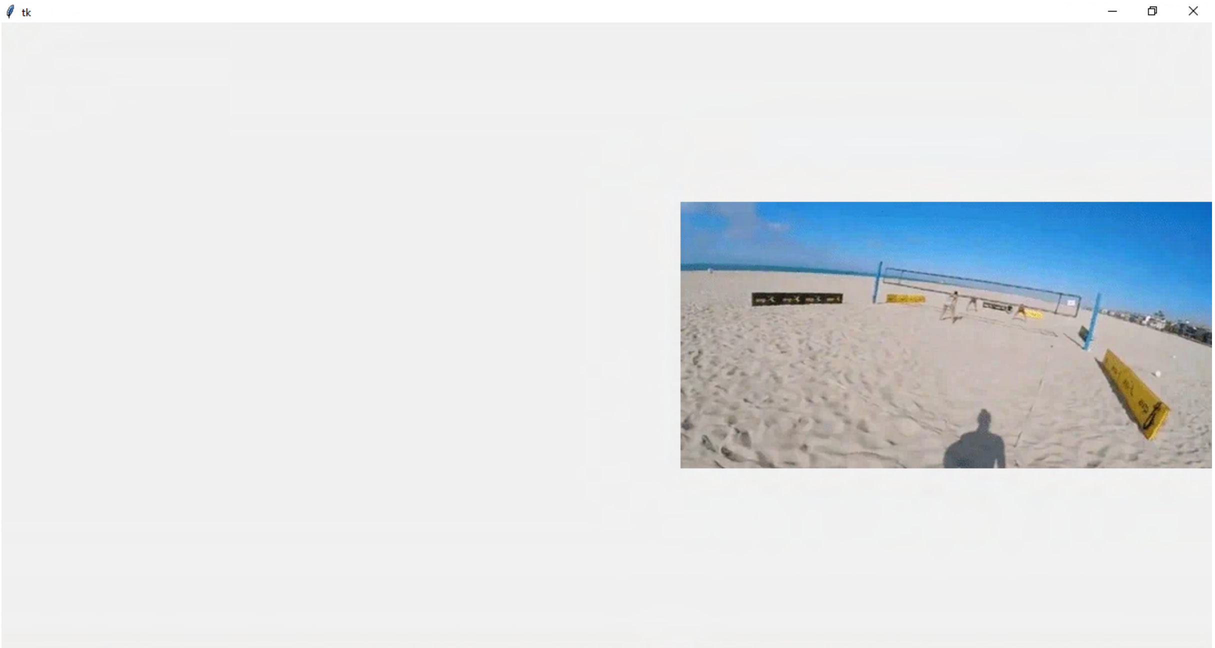

Fp1
Fp2
C3
1e−8
C4
P7
P8
O1
O2
forward
right
backward
left
stop
Direction

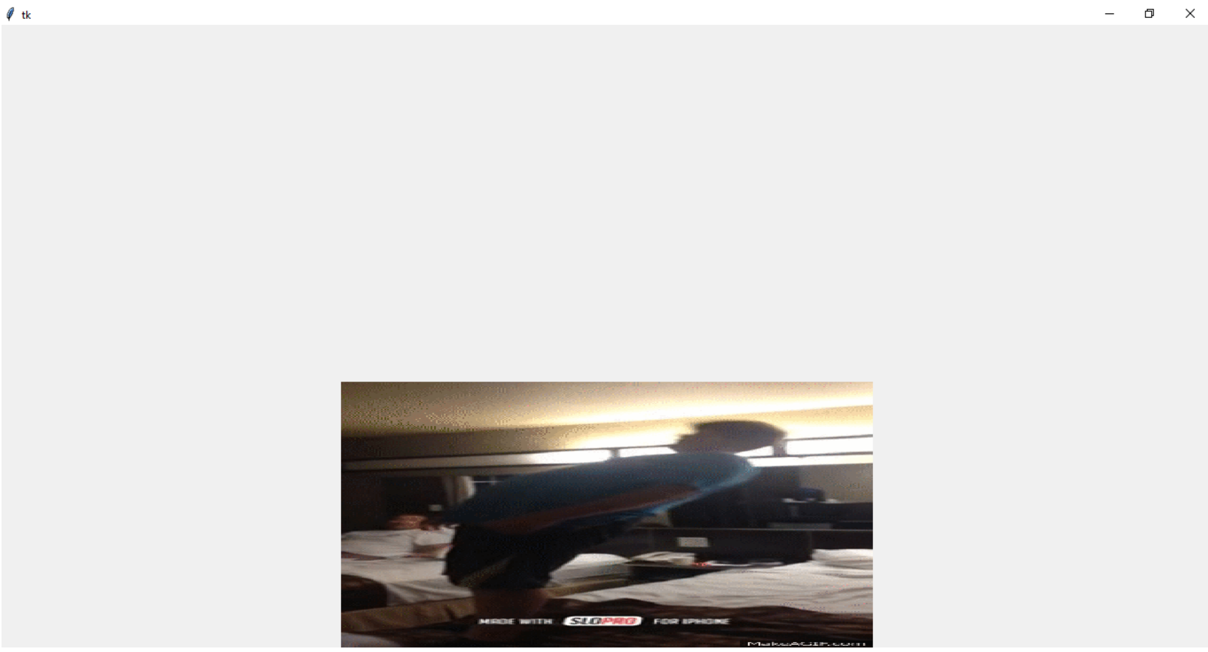

F7
F8
F3
F4
T7
T8
P3
P4
forward
right
backward
left
stop
Direction

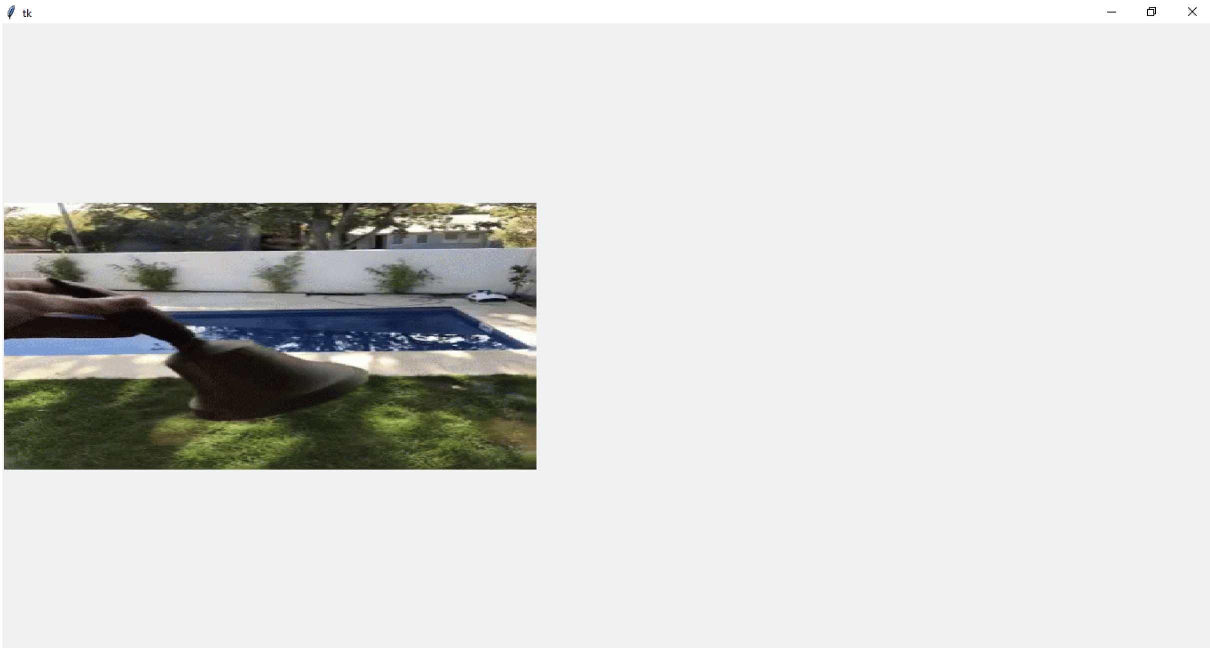

tk
NEXT
10 (~5 min)
15 (~7.5 min)
20 (~10 min)

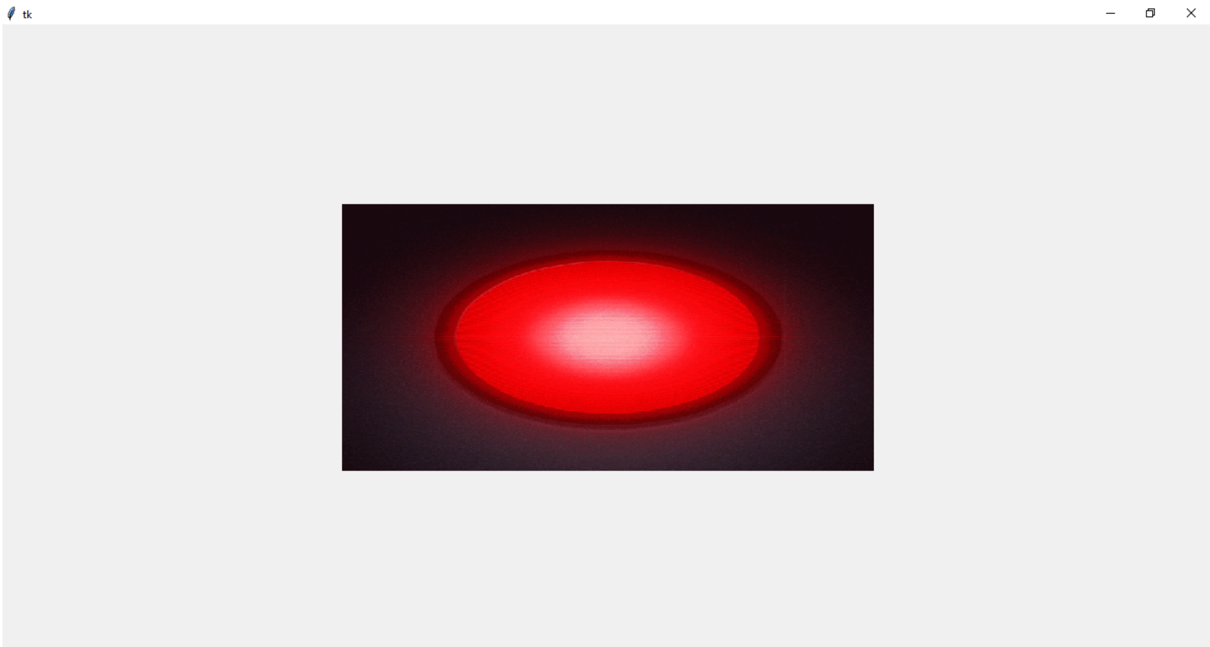

tk
START

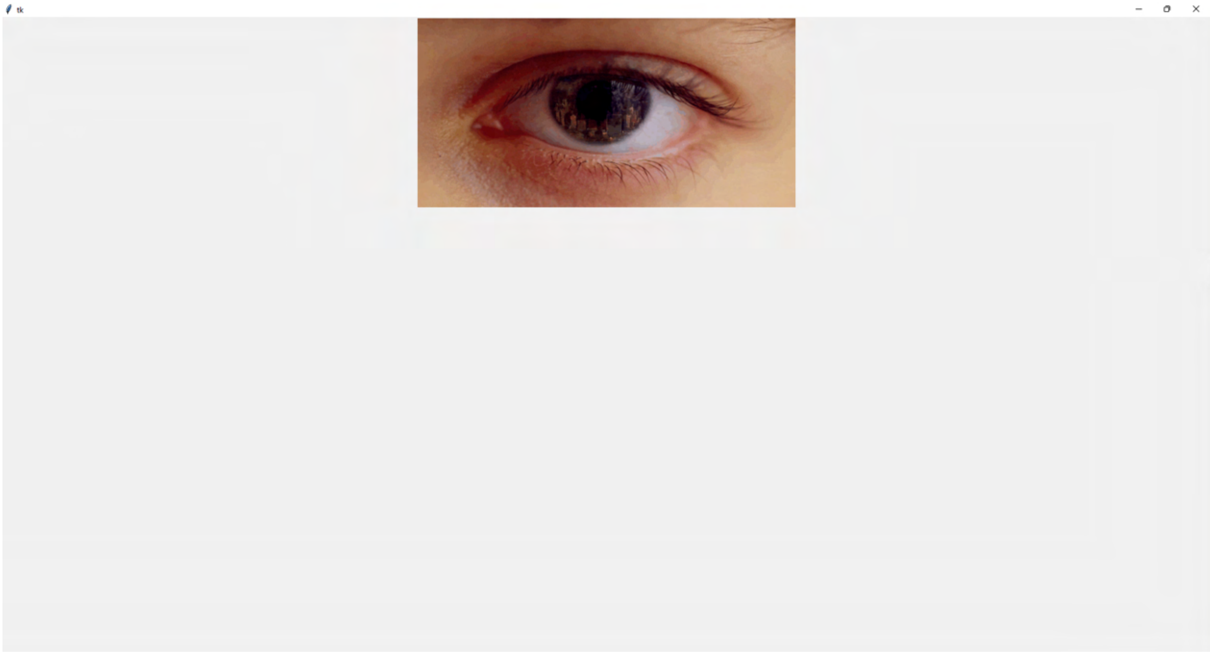

tk

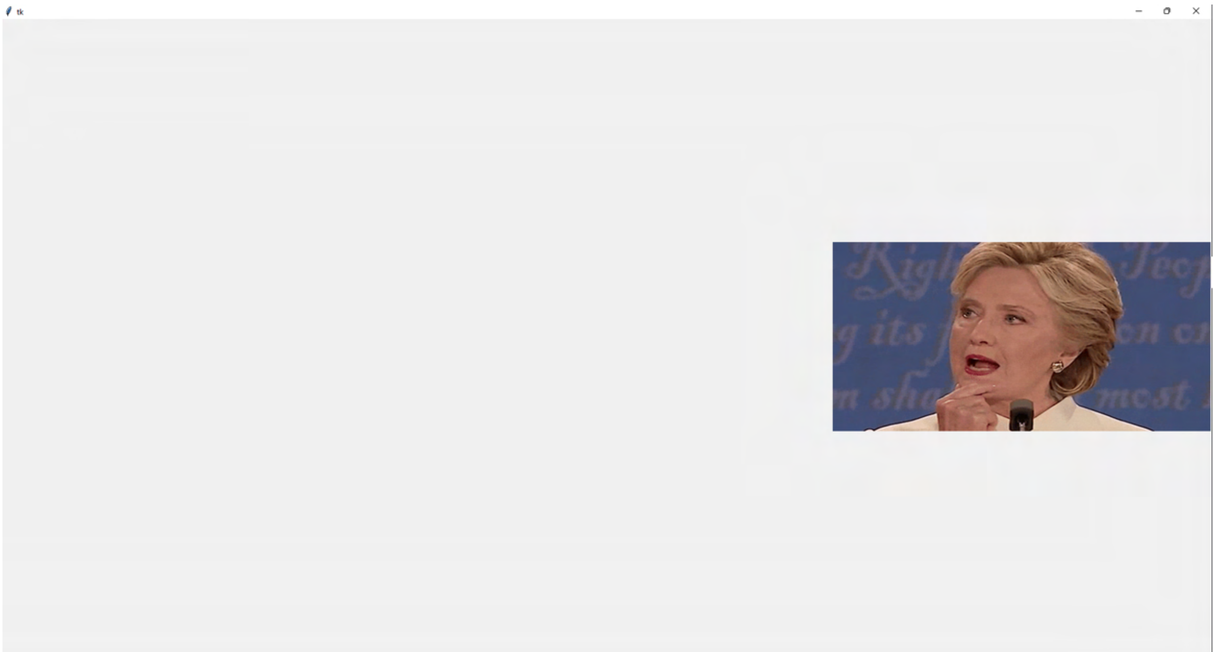

tk

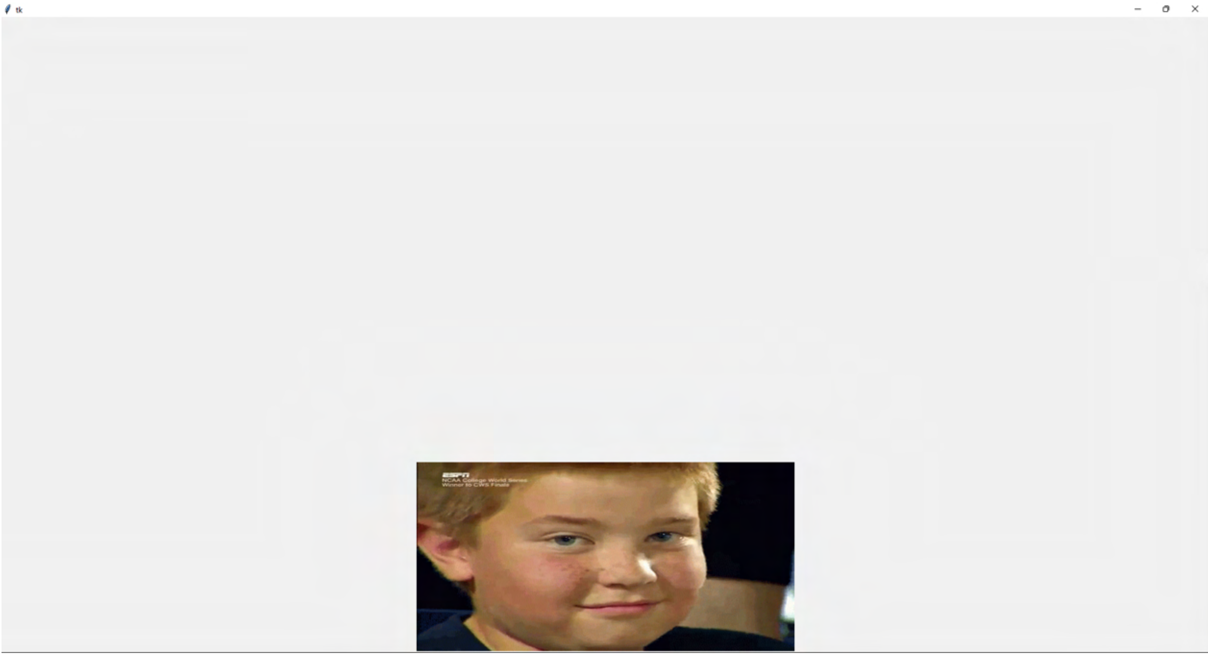

tk

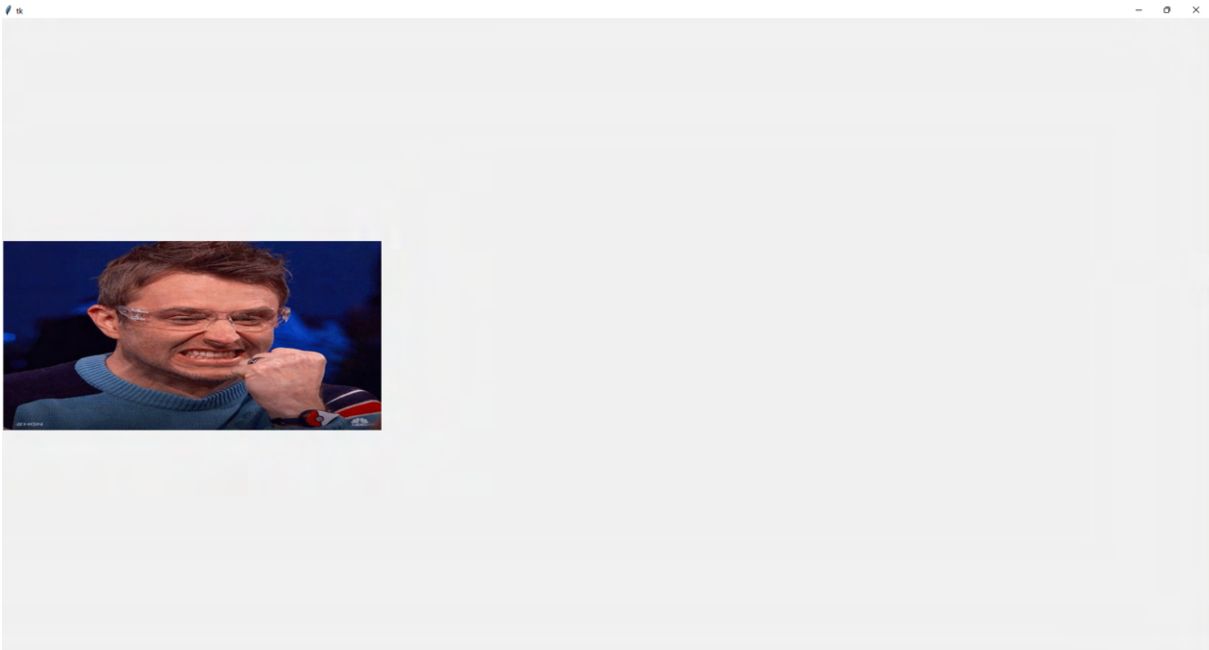

tk

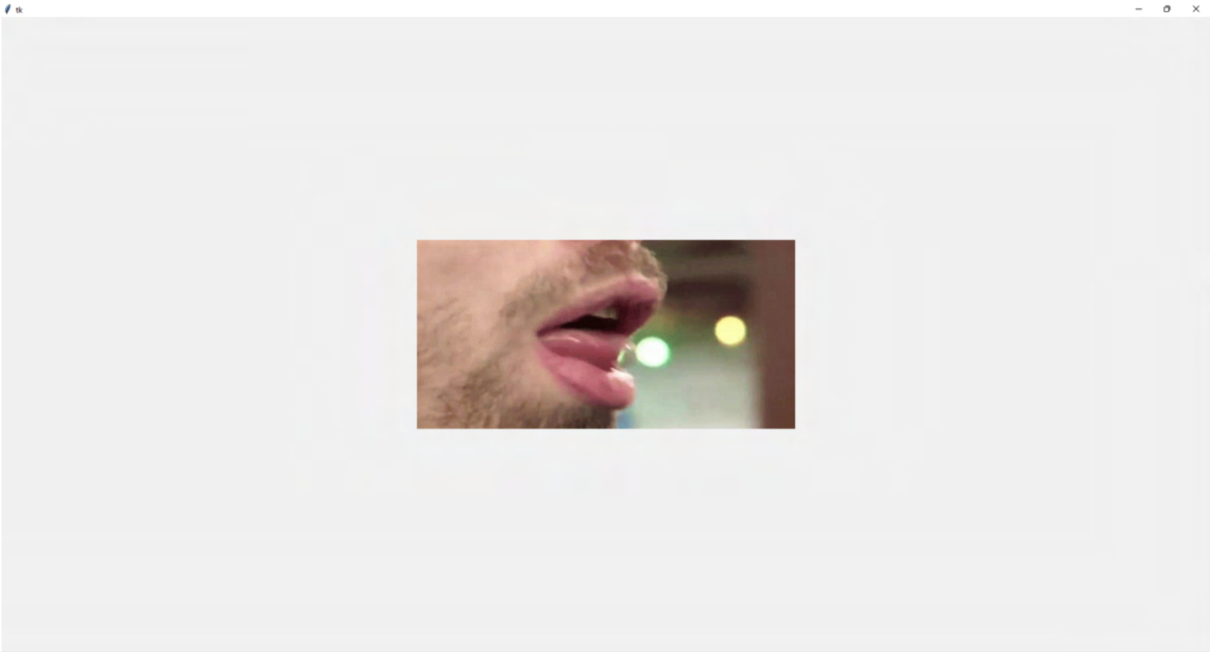

tk

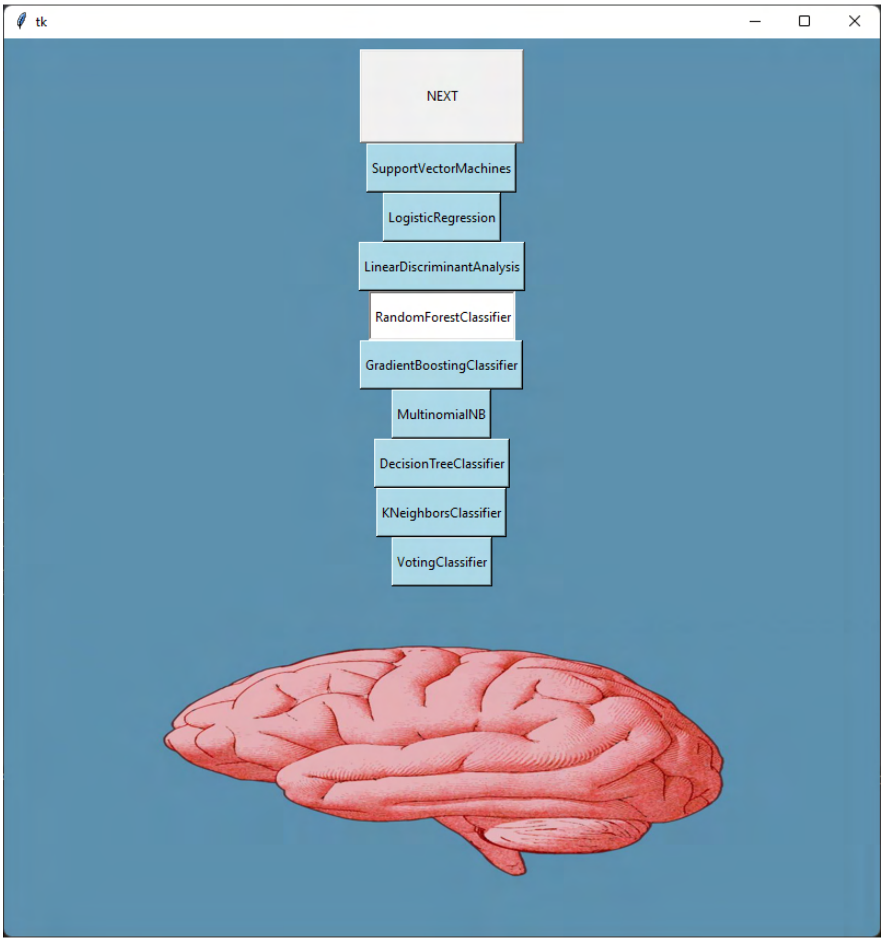





tk


tk


tk


tk
NEXT
SupportVectorMachines
LogisticRegression
LinearDiscriminantAnalysis
RandomForestClassifier
GradientBoostingClassifier
MultinomialNB
DecisionTreeClassifier
KNeighborsClassifier
VotingClassifier

REFERENCES

[1] I.A. Otto, M. Kon, A.H. Schuurman, L.P. van Minnen, Replantation versus prosthetic fitting in traumatic arm amputations: a systematic review, PloS One 10 (9) (2015) e0137729.

[2] M. C. Kiernan, S. Vucic, B. C. Cheah, M. R. Turner, A. Eisen, O. Hardiman, J. R. Burrell, and M. C. Zoing, “Amyotrophic lateral sclerosis,” The Lancet, vol. 377, no. 9769, pp. 942–955, 2011. [Online].
Available: https://www.sciencedirect.com/science/article/
pii/S0140673610611567

[3] O. Barron, M. Raison, and S. Achiche, “Control of transhumeral prostheses based on electromyography pattern recognition: From amputees to Deep Learning,” Powered Prostheses, 17-Apr-2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128174500000018. [Accessed: 13-Nov-2021].

[4] A. Miskon, S. Thanakodi, S. M. H. Azhar, and A. K. S. Djonhari, “Identification of raw EEG signal for prosthetic hand ...,” ResearchGate, Dec-2019. [Online]. Available: https://www.researchgate.net/publication/341123944_Identification_of_Raw_EEG_Signal_for_Prosthetic_Hand_Application. [Accessed: 13-Nov-2021].

[5] Li R, Zhang X, Lu Z, et al. An Approach for Brain-Controlled Prostheses Based on a Facial Expression Paradigm. Front Neurosci. 2018;12:943. Published 2018 Dec 18. doi:10.3389/fnins.2018.00943 An Approach for Brain-Controlled Prostheses Based on a Facial Expression Paradigm (nih.gov)

[6] Nsugbe, E.; Samuel, O. A Self-Learning Control Scheme for Upper-Limb Prosthesis Control Using Combined Neuromuscular and Brain Wave Signals, in Proceedings of the 7th International Electronic Conference on Sensors and Applications, 15–30 November 2020, MDPI: Basel, Switzerland, doi:10.3390/ecsa-7-08169, Available: https://sciforum.net/paper/view/8169

[7] Y. Jiang, C. Chen, X. Zhang, W. Zhou, C. Chen, en S. Lemos, “EEG-Based Hand Motion Pattern Recognition Using Deep Learning Network Algorithms”, in Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition, Xiamen, China, 2020, bll 73–79.

[8] S. Gannouni, K. Belwafi, H. Aboalsamh, Z. AlSamhan, B. Alebdi, Y. Almassad, and H. Alobaedallah, "EEG-Based BCI System to Detect Fingers Movements," *Brain Sciences*, vol. 10, no. 12, p. 965, Dec. 2020.

[9] Turnip, Arjon & Pardede, Jasman. (2017). Artefacts Removal of EEG Signals with Wavelet Denoising. MATEC Web of Conferences. 135. 00058. 10.1051/matecconf/201713500058.