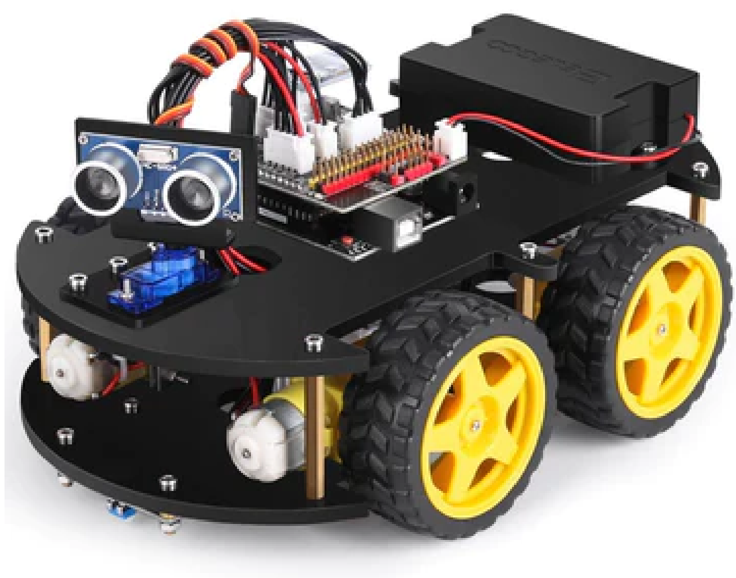
ThoughtRobot Documentation

Domenic Rothenberger 78158,
Ali Karami 80018,
Roman Dietenmeier 79709,
Matthias Pijarowski 79673

January 2023



Contents

1 Introduction **3**

2 Project overview and goal **3**
2.1 Goals . . . 3
2.2 High-level view . . . 4
2.3 Features and Capabilities . . . 4

3 Description of functionality and scenarios **5**
3.1 Command receiver structure . . . 5
3.1.1 Robot-Gateway Receiver Workflow . . . 5
3.2 Ability to control the robot through hand gestures using hand recognition technology . . . 6
3.3 Ability to control the robot through voice commands using Alexa 7
3.4 Ability to control the robot through brain signals using Brain-Computer-Interface (BCI) technology . . . 7
3.5 Ability to record and replay patterns . . . 8

4 Architectural description **10**
4.1 Gesture Classifier . . . 10
4.2 Alexa . . . 11
4.3 BCI . . . 12
4.4 Class diagram . . . 14

5 Reused **15**
5.1 Reused Hardware . . . 15
5.1.1 Elegoo Smart Robot Car Kit V4.0 . . . 15
5.1.2 Alexa . . . 15
5.1.3 EMOTIV EPOC X . . . 16
5.1.4 Raspberry Pi 4 . . . 17
5.1.5 Camera . . . 17
5.2 Reused Software . . . 17
5.2.1 Elegoo Smart Robot Car Kit V4.0 . . . 17
5.2.2 EMOTIV Launcher and EMOTIV BCI . . . 18

6 Description of additional functionality **18**
6.1 Elegoo Smart Robot Car Kit V4.0 . . . 18

7 Lessons Learned **19**

8 Appendix **19**

1 Introduction

The ThoughtRobot is a versatile and innovative robot that employs a variety of cutting-edge technologies to provide an intuitive and user-friendly experience. The robot can be moved using a Brain-Computer Interface (BCI), as well as voice commands via Alexa and hand recognition techniques. This documentation will provide a detailed overview of ThoughtRobot's features and capabilities, as well as instructions for setting up and operating the robot. We hope that this documentation will provide you with the information you need to fully utilize and appreciate the capabilities of ThoughtRobot.

2 Project overview and goal

2.1 Goals

The ThoughtRobot project's main goal is to create a robot that can be controlled using a variety of methods, including hand gestures, voice commands, and brain signals. This goal emphasizes the robot's versatility and flexibility, allowing users to interact with it in the most convenient or natural way for them.

The project's goal is to develop a user-friendly interface for interacting with the robot, allowing users to easily operate the robot and access its features by utilizing the three controlling features: hand gestures, voice commands, and brain signals.

Another goal is to show how cutting-edge technologies like Brain-Computer Interface (BCI) and Alexa voice commands can be integrated into a real-world application, providing a glimpse into the future potential of these technologies.

The project also intends to serve as a platform for future research and development in the field of mobile and embedded systems.

Overall, the ThoughtRobot project's main goal is to create a robot that can be controlled in a variety of ways, providing users with a versatile and intuitive experience while also demonstrating the potential of advanced technologies in real-world applications.

2.2 High-level view

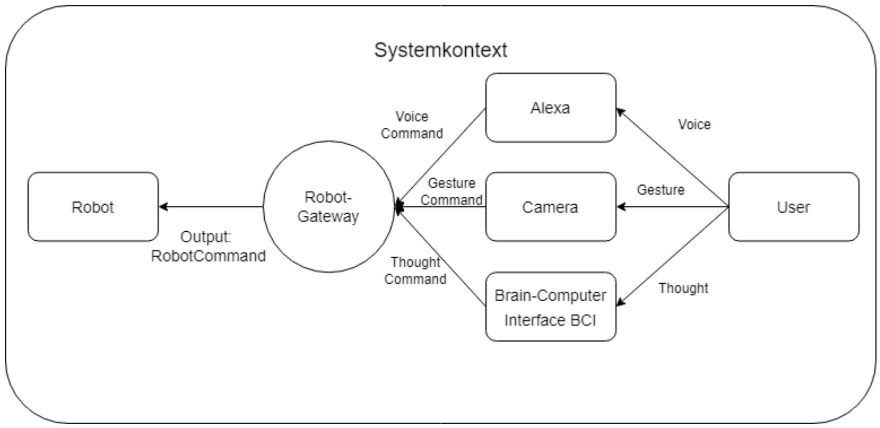


Figure 1: System Behaviour with single movement intention

Human Hands, Alexa, and Brain-Computer Interface(BCI) are the system's three actors. "Control Robot" would be the main use case, with three sub-use cases: "Control with Hands," "Control with Voice," and "Control with BCI." Each sub-use case would represent a different way to interact with the robot, with "Control with Hands" representing the ability to control the robot using hand gestures, "Control with Voice" representing the ability to control the robot using voice commands through Alexa, and "Control with BCI" representing the ability to control the robot using brain signals via BCI. Each use case has a different set of steps or actions that the system needs to perform in order to complete the use case. The actors "Human Hands," "Alexa," and "BCI" would be linked to the main use case "Control Robot" by a line with an arrowhead pointing to the use case, indicating that the actors are interacting with the system to complete the use case. Figure 1 is a high-level view of the ThoughtRobot projects use case diagram.

2.3 Features and Capabilities

Here is a list of the ThoughtRobot´s features and capabilities:

- Ability to control the robot through hand gestures using hand recognition technology.
- Ability to control the robot through voice commands using Alexa.
- Ability to control the robot through brain signals using Brain-Computer Interface (BCI) technology.
- Ability to record and save routines or patterns.

3 Description of functionality and scenarios

This section describes the overall functionalities of the system and how the different commands are created. An in detail description of the inputs registered by the different commands is provided in the appendix. The general process of the system's behavior after receiving a movement command is described in the sequence diagram below.

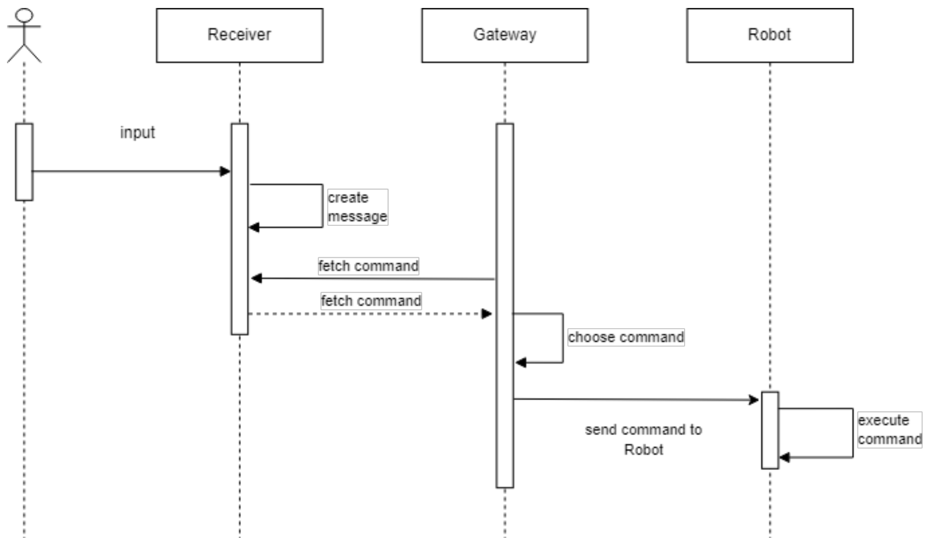


Figure 2: System Behaviour with single movement intention

3.1 Command receiver structure

The system is designed to allow any form of input as long as this input is received within a receiver implementing the specified "ReceiverInterface". Additionally, any Receiver needs to be registered in the Gateway. Internally the system provides a finite number of inputs that can be processed. Ideally, the received inputs have an equivalent layout to the system's internal commands. In the next passage, this workflow is described in detail.

3.1.1 Robot-Gateway Receiver Workflow

The internal System uses the following commands to represent the different operations.

- Operations regarding the Recording of a Pattern:
  - StartRecording
  - StopRecording

- Name / Symbol a Pattern can be mapped to:
  - Patternone
- Robot Single Movement Actions:
  - stop
  - forward
  - backwards
  - right
  - left

Each Receiver class can thus use these operations to push them forward to the Robot. If additional functionality or keywords are desired the Configuration of the System needs to be adjusted.
We as a team then created Receivers for the different technologies needed for every input method, which we discuss in the following sections. If you wish you could even add more Receivers to the project, as they are all easily interchangeable as mentioned.

3.2 Ability to control the robot through hand gestures using hand recognition technology

The "Ability to control the robot through hand gestures using hand recognition technology" feature enables users to control the Thought Robot by performing specific hand gestures. Because hand gestures are a common and familiar form of communication, this provides an intuitive and natural way for users to interact with the robot.

This feature's hand recognition technology analyzes the shape and movement of the user's hand and matches it to a predefined set of gestures. Pointing the index finger up, for example, indicates that the robot should move forward. Whereas an open hand with the palm facing up indicates that the robot should stop moving. This allows users to control the robot in a manner similar to that of a remote-controlled car.

To execute and record a routine or a pattern as mentioned in the section 2.3 the system provides a set of hand gestures to do that. We decided to exclusively offer this capability using hand gestures since this form of input turned out to be the most reliable. In the sub-section 3.5 the process of executing a pattern is further described.

In regards to the usability of the robot in our everyday life, we picture the following scenarios.
In a warehouse where the robot is used to move goods, the operator can control the robot with hand gestures without the need for a controller or a device, moving the robot forward and backwards, turning left and right, and stopping it with simple hand gestures. This feature is also useful when the operator's hands are busy or dirty and they cannot use a controller or a device, such as

when the operator is working on a machine and needs to control the robot with one hand.

Overall, the ability to manage the robot via hand gestures using hand recognition technology provides a comfortable and intuitive approach for users to interact with the robot, making it simple to operate and access its functions.

3.3 Ability to control the robot through voice commands using Alexa

The ”Ability to operate the robot by voice commands using Alexa” feature enables users to control the ThoughtRobot by speaking specific commands to an Alexa-enabled device. Because voice instructions are a natural and familiar method of communication, this provides a convenient and easy approach for users to communicate with the robot.

This feature’s Alexa-enabled gadget operates by listening to the user’s voice and interpreting the uttered command. A user may, for example, say ”go forward” to cause the robot to move forward, or ”stop” to for the robot to cease moving.

In a home situation when the robot is being used as a personal assistant, this function allows the user to control the robot using voice commands without the need for a controller or a tablet. The user can instruct the robot to drive to any location. This feature is especially useful for persons who have mobility challenges or when the user’s hands are full.

Overall, the ability to manage the robot via Alexa voice commands provides a straightforward and intuitive approach for users to engage with the robot, making it simple to operate and access its functions. The user can deliver commands to the robot in a natural and familiar manner, and the robot can respond in the same manner.

3.4 Ability to control the robot through brain signals using Brain-Computer-Interface (BCI) technology

The ”Ability to operate the robot by brain signals using Brain-Computer Interface (BCI) technology” feature enables users to control the ThoughtRobot by transmitting precise signals from their brains. Because brain signals are a direct and natural type of communication, this enables a unique and advanced approach for users to connect with the robot.

This feature’s Brain-Computer Interface (BCI) technology records the electrical activity of the user’s brain, often via electrodes inserted on the scalp. We use the Emotiv Epoc X BCI headset. The impulses received from the EEG (electroencephalogram) are subsequently translated into directives that govern the motions of the robot. A user could, for example, consider moving their hand forward to have the robot move forward, or stopping to make the robot stop moving.

This function could be useful at a rehabilitation center where the robot is used to assist patients with movement impairments. Patients can control the robot with their thoughts, allowing them to do physical therapy exercises and regain movement in a more natural manner without the use of controllers or equipment. It can also be used when the user is unable to move or talk, such as when suffering from paralysis or locked-in syndrome.

Overall, the ability to control the robot via brain signals via BCI technology provides a novel and advanced manner for users to interact with the robot, making it simple to operate and access its functions. The user can deliver commands to the robot in a natural and direct manner, which is very useful for persons who have mobility challenges or are unable to move or talk. We achieved to move the robot by reading brain waves with the Emotiv Epoc X BCI headset. However, it should be noted that BCI technology is still in its early stages and may necessitate additional equipment and setup.

3.5 Ability to record and replay patterns

In Addition to receiving and executing movement commands from the user, the robot possesses the ability to record successive movement commands and replay these patterns on command. The following illustrations demonstrate the system's behavior in either of the two cases.

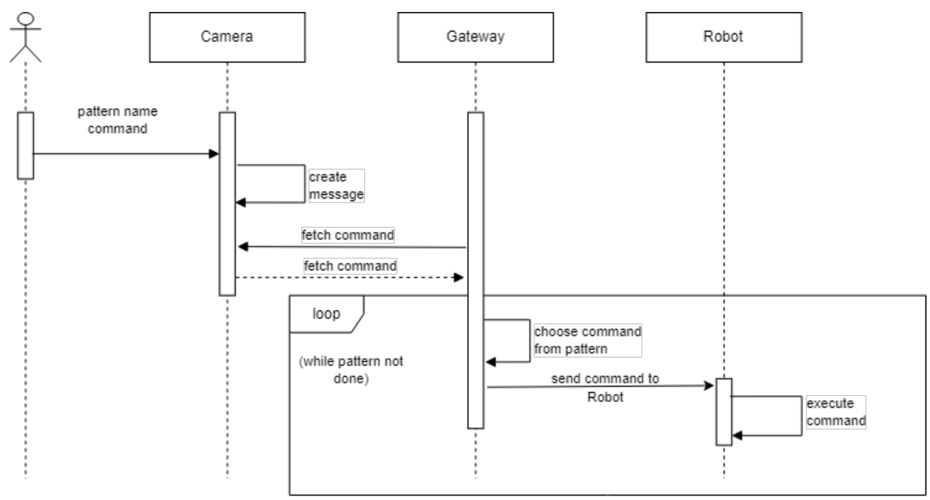


Figure 3: System Behaviour with pattern executing intention

In figure 3 you can see how the system runs a routine after an actor has initiated a pattern command.

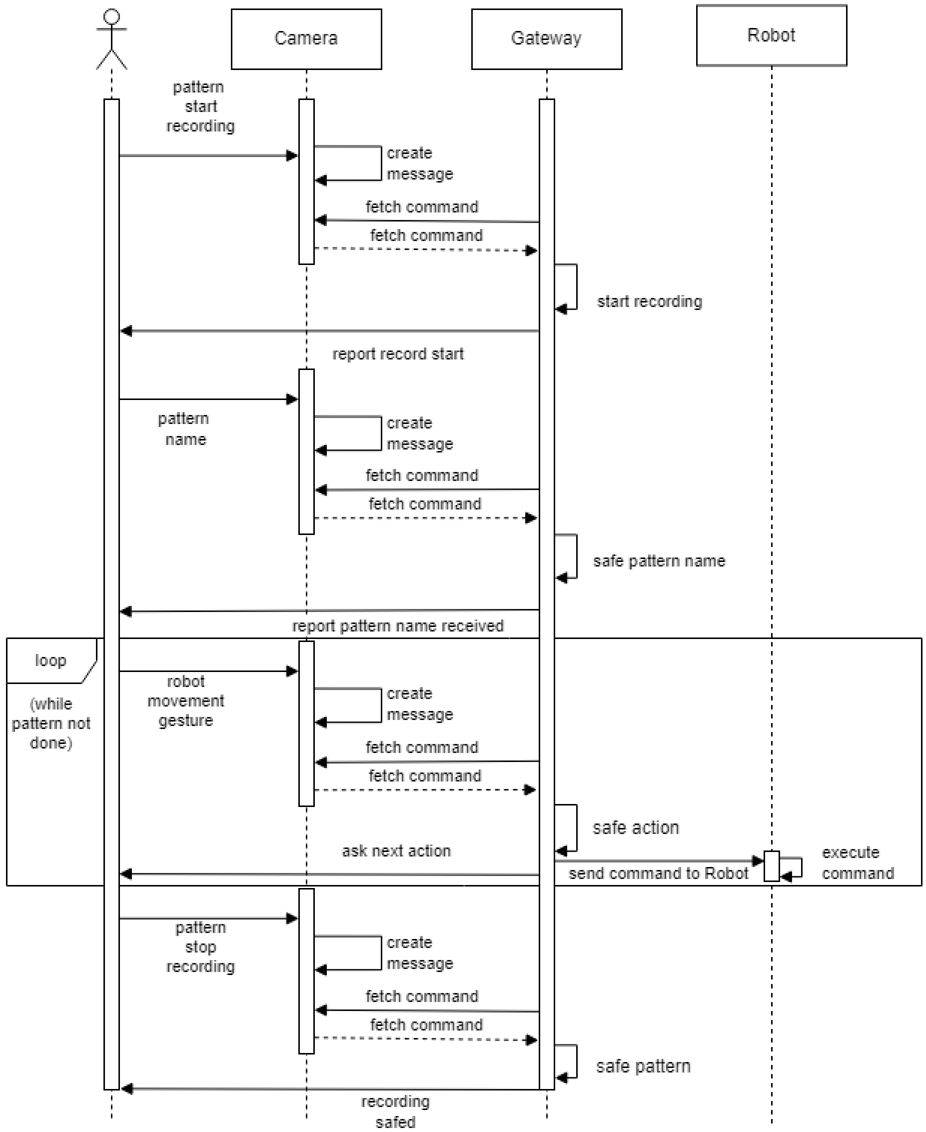


Figure 4: System Behaviour with pattern recording intention

And in figure 4 you can see how an actor creates a new routine.

4 Architectural description

4.1 Gesture Classifier

The gesture classifier is built as a three-stage classifier. The first two stages are a region proposal network, which extracts the hand from an image. This cropped image is then further processed by classifying and localizing the key-points of the hand. For the first two stages we decided to use the mediapipe library from google https://google.github.io/mediapipe/solutions/hands.html. For the third stage we used a 1d-convolution network, which we implemented with the popular library "pytorch" https://pytorch.org/. The structure can be seen in figure 5. It consists of three convolutional layers to extract features of the keypoint vector and combines those into more complex features. In the end, it then uses two linear layers to combine the output of the last convolution layer and outputs either logits or probabilities for each hand gesture. This model was then trained on a dataset we recorded ourself. It contains eighth gestures. Training was done using the pytorch-model-trainer library https://pypi.org/project/pytorch-model-trainer/. The gestureClassifier class takes care of those steps and maps the output of the classifier to the gesture with the highest probability. The process of classifying an image can be seen in figure 6.

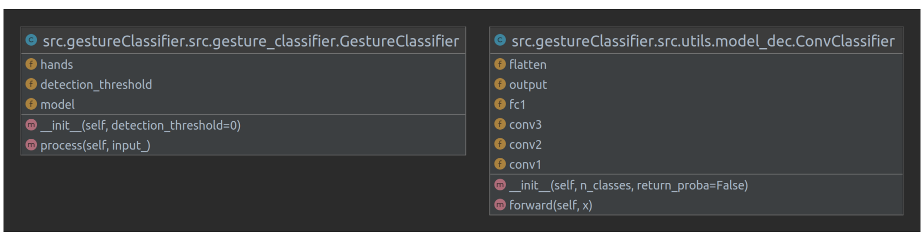


Figure 5: Gesture Classifier class diagram

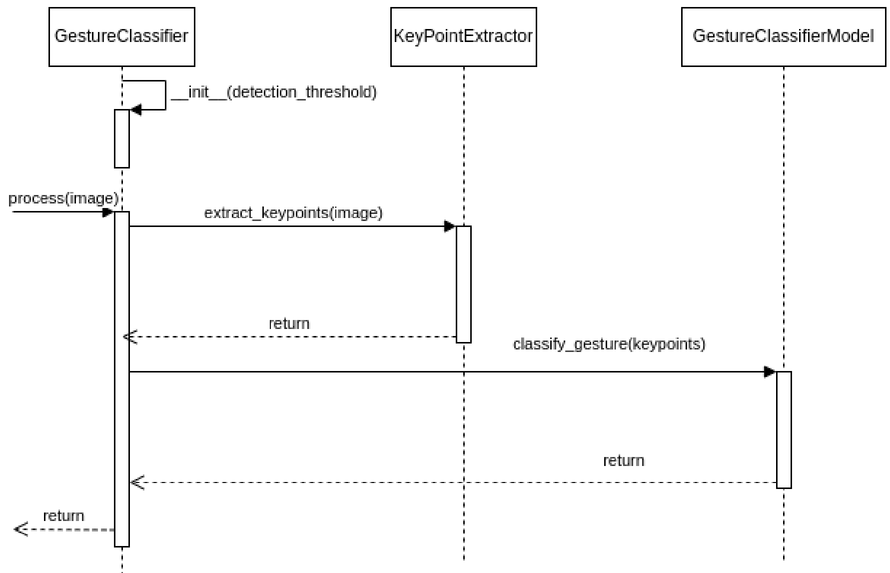


Figure 6: Gesture Classifier sequence diagram

4.2 Alexa

For our Alexa skill we used the Alexa developer console. We added a custom intent to the preexisting ones. The intent is called RobotControlIntent and can recognize common phrases for how a user would interact with the robot. You can for example say phrases like "Fahre nach ⟨direction⟩". For ⟨direction⟩we also build a custom slot type called "Direction". It can have the predefined values "Anhalten", "Rückwärts", "Vorwärts", "Links" and "Rechts". In the sequence diagram 7 you can see that when the RobotControlIntent is called, it will extract the word with the highest probability. This word is then published to our MQTT server, which our Raspberry Pi subscribes to, to mediate the commands forward to the Robot.

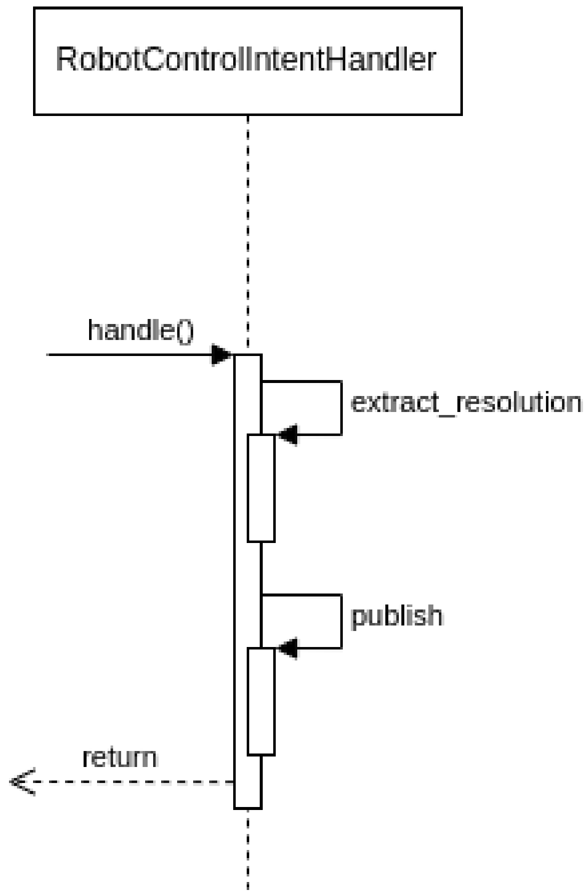


Figure 7: Alexa sequence diagram

4.3 BCI

For BCI, we use an MQTT server and the Emotiv API to read commands from an Emotiv headset via a socket connection. These commands are then sent to our MQTT server. The robot listens to this topic through a command handler, extracts the command, and maps it to the appropriate action on the robot.

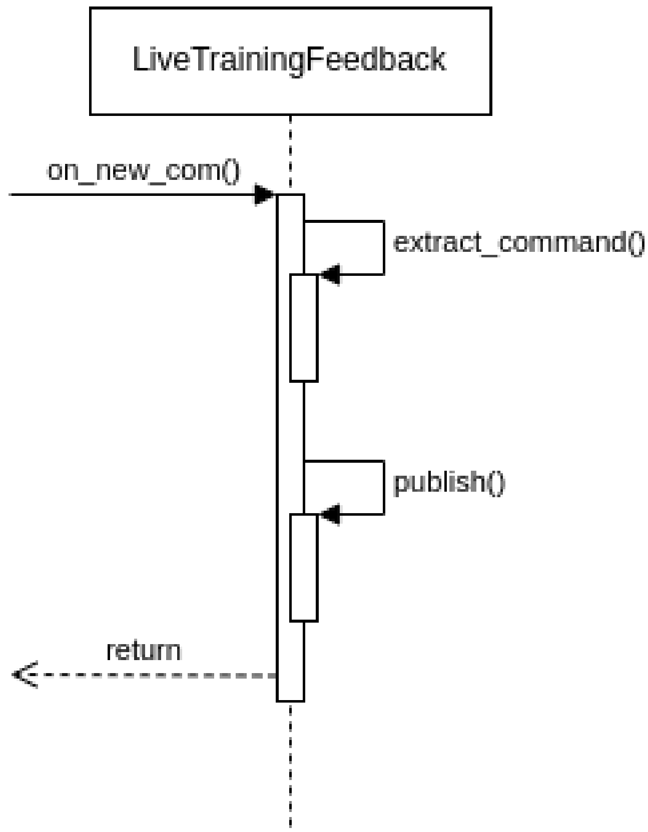


Figure 8: BCI sequence diagram

4.4 Class diagram

The following class diagram describes the system's internal classes and associations between them.

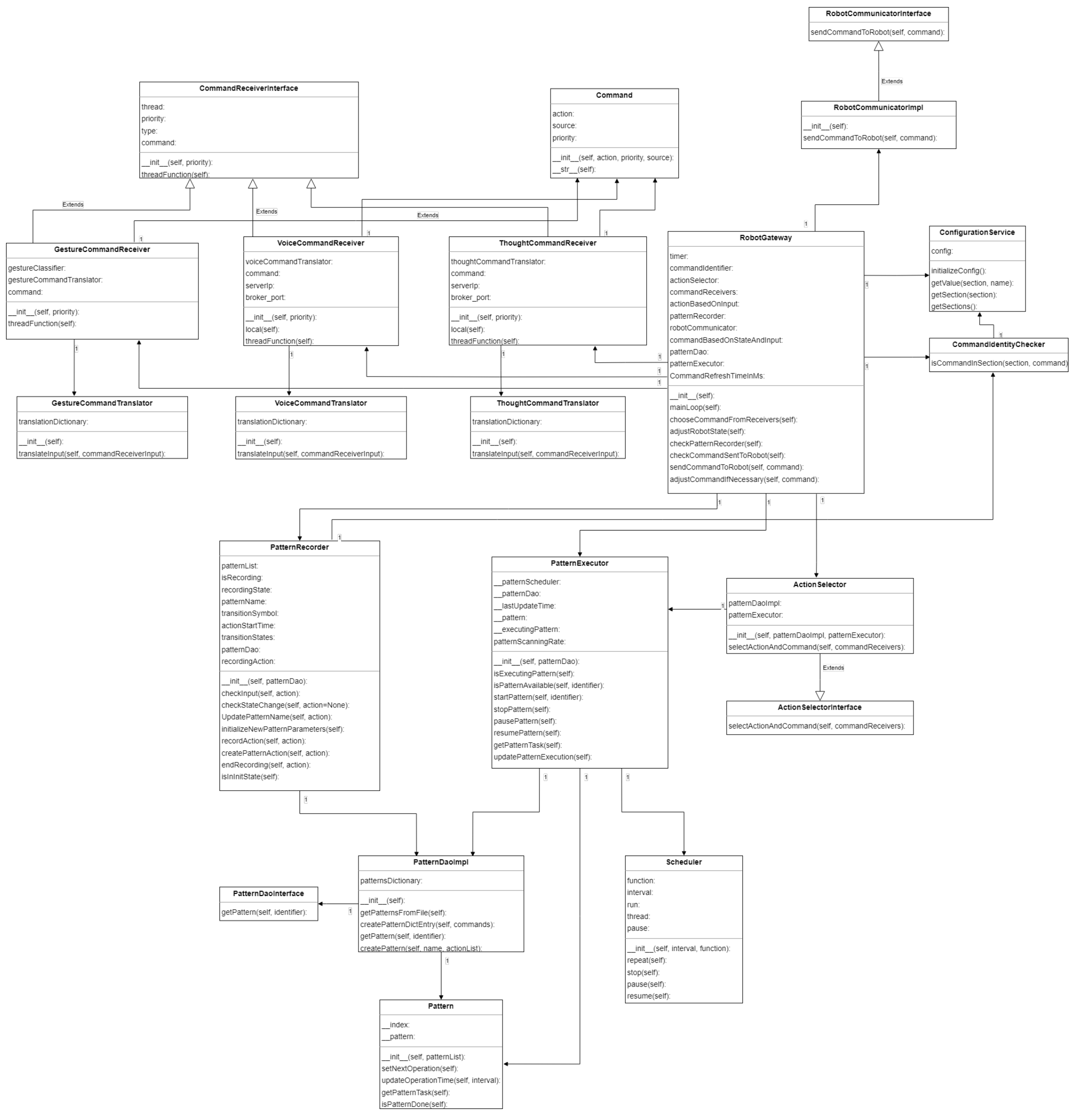


Figure 9: class diagram

The class diagram showcases the whole RobotGateway with its Receivers and Pattern or Routine executor, which runs on the Raspberry Pi 4.

5 Reused

As our project was focused on robot input methods, we did not try to build a new robot ourselves, but the tools necessary to use different new types of inputs to control a robot.

5.1 Reused Hardware

So we reused a Robot, an Alexa, a Camera, a BCI Headset as well as a Raspberry Pi 4 to mediate between the different gadgets.

5.1.1 Elegoo Smart Robot Car Kit V4.0

The "Elegoo Smart Robot" comes with many possible ways to control the Robot himself. He supports out of the box control via:

- infrared remote
- Handy App over WiFi

This is possible due to the built-in Arduino Uno plus shield with an infrared receiver. As well as a built-in ESP32 Microcontroller which is responsible for the WiFi connection.
Later in our project, we made great use of the ESP's WiFi features. As this became the entry point for all the commands mediated by the Raspberry Pi.

Figure 10: Elegoo Smart Robot Car Kit V4.0

5.1.2 Alexa

As mentioned before in section 3.3, we used Amazon Alexa to provide a speech to text interface to control the Thoughtrobot via voice command.

Figure 11: Alexa

5.1.3 EMOTIV EPOC X

The Emotiv Epoc X headset is a brain-computer interface (BCI) device that uses 14 dry electrodes to track and measure brain activity. The headset is known for its high-quality, professional EEG technology and can be used in a variety of research applications, including neurofeedback, brain-computer interfaces, and cognitive assessment.

A key advantage of the Emotiv Epoc X headset is the high resolution of its EEG data, which enables precise measurements of neural activity. The device also has a lightweight and practical design that can be used for long periods in research experiments. The Epoc X also has software tools, including EmotivPRO Research software, which provides a variety of features and visualization options for data analysis.

Figure 12: EMOTIV EPOC X

5.1.4 Raspberry Pi 4

A Raspberry Pi is a small full functioning computer and thus perfectly suited to assist our Robot, as you could later attach the Raspberry Pi to the Robot to have the new features without being bound to a specific location.
We used the Raspberry Pi for the following:

- to manage all the commands the Robot should execute
- to receive MQTT messages from the Alexa or the BCI Headset
- to process camera input
- to save routines and for all the logic of handling all the different input methods

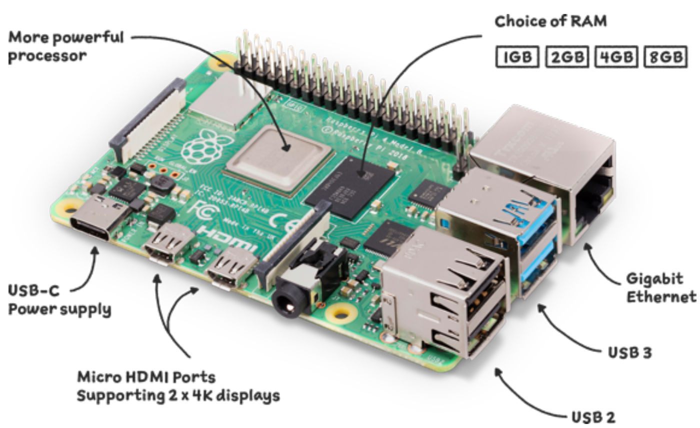


Figure 13: Raspberry Pi 4

5.1.5 Camera

We attached a regular USB-Webcam to the Raspberry Pi so that the Raspberry can have a static camera input source instead of using the camera attached to the Robot itself.

5.2 Reused Software

The Robot and the BCI Headset came with preexisting Software to use. Which reduced our development times as we again could focus on implementing new input methods.

5.2.1 Elegoo Smart Robot Car Kit V4.0

The ESP Microcontroller built into the Robot already came with a full communication Schema to use over a TCP connection. So we could send the preexisting commands from the Raspberry Pi to the Robot without much more complication.

5.2.2 EMOTIV Launcher and EMOTIV BCI

To set up the system, we first connected the Emotiv headset to our computer using the Emotiv Launcher software. This software allows us to easily establish a connection between the headset and our computer, and also provides us with useful information about the headset's status and battery level.

Once the connection was established, we moved on to the next step, which was to use the Emotiv BCI software to train our model. This software allows us to create a user profile, which is essentially a set of data that represents an individual's brainwaves. We then used this profile to train the model so that it could recognize and respond to different mental states and commands.

After the training was completed, we were able to use the SDK and socket connection to interact with the model. This allowed us to control various devices and perform different actions based on the command that the model recognized. It was a very exciting experience to see the model respond to our thoughts and see the potential applications of BCI technology.

6 Description of additional functionality

As we reused the "Smart Robot," we still made some adjustments.

6.1 Elegoo Smart Robot Car Kit V4.0

The ESP32 Microcontroller already created its own WiFi Network, but this was not sufficient as this network would not provide access to the internet. Thus we made adjustments to the source code of the ESP32, so that the Microcontroller would not only create its own network but also connect to another WiFi. This way the Robot could also be reached in a network with internet access, which allowed the Raspberry Pi to receive Messages from a MQTT Server.

7 Lessons Learned

- Understanding of computer vision, including image processing technologies and how to use them to detect and recognize hand gestures.
- The use of artificial intelligence and machine learning techniques to increase the intelligence of a robot.
- We learned about Brain-Computer Interface (BCI) and how to interpret brain signals with an EEG headset.
- Knowledge of how Alexa works and how to create Alexa apps.
- The ability to integrate BCI, EEG, and Alexa into the control system of a robot.
- Understanding and extending code from unknown Developers from a different culture. As we modified the code of the Robot.
- Outstanding project management, planning, organization, collaboration, and communication abilities.

8 Appendix

The appendix starts off with a chronological listing of all sprint report documents. Right after that follows a document that describes the different commands that we defined as part of the project.

- CONOPS MS Report
- Tech Demo MS Report
- Architectural Spike MS Report
- Alpha MS Report
- Beta MS Report
- Implemented Gesture Input

CONOPS MS Report

Domenic Rothenberger 78158,
Ali Karami 80018,
Roman Dietenmeier 79709,
Matthias Pijarowski 79673

October 2022

1 Introduction

Within our Project, we plan on combining Robot technology with an brain-computer-interface (BCI). Brain-computer interfaces allow users to interact with a system through mental actions alone, as opposed to traditional control methods such as physical manipulation or verbal commands. Ideally the user should be able to fully control the robot over said interface.

Addition from 27.10.22:

The actions of the robot are controlled via voice recognition or hand-gestures. In addition to the robot directly performing one action in response to one user input the software should be able to perform routines (sequences of actions). These actions will be launched via a keyword or gesture specific gesture.
Optionally the system can learn new routines through its input peripherie.

2 Scenarios

The main scenarios are described in the form of Use-Cases.

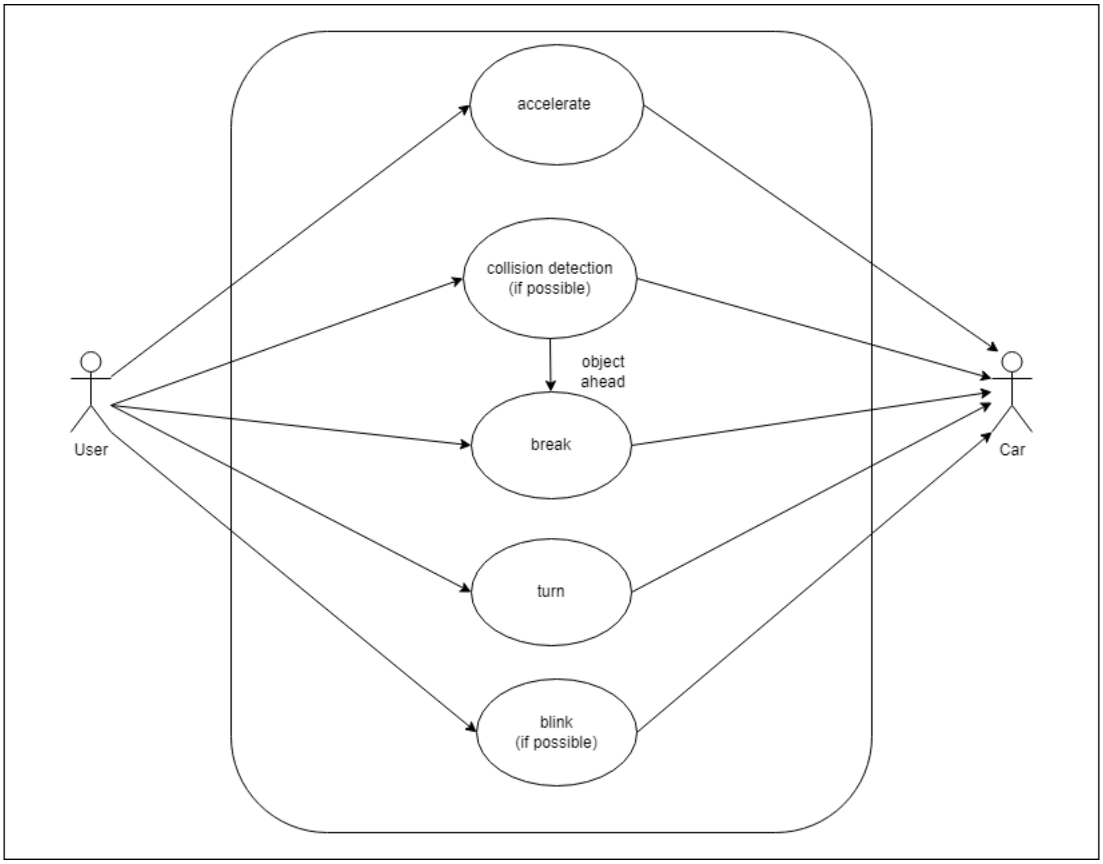


Figure 1: Use-Cases

3 Techology choices

The signals from the EEG headset are sent to the computer.
A Python programm then interprets the received signals and determines the corresponding action.
The rasperrypi communicates with the robot and checks whether the person is in focus, and if so, the robot moves.
As of now we plan on using one of the following robots depending on Stakeholder approval.

- Amazon car1
- Amazon car2
- Amazon car3

Each of these cars is then controlled via a Arduino. The functionality is written with python.
Further interpreting of the voice or visual commands is performed by a central backend sitting on a additional computer or RaspberriPi depending on the needed resources.
The voice commands are recognized with an alexa.
The handgestures are recorded with a webcam connected connected to the pc or RaspberryPi.

Tech Demo MS Report

Domenic Rothenberger 78158,
Ali Karami 80018,
Roman Dietenmeier 79709,
Matthias Pijarowski 79673

November 2022

During this sprint we ordered our robot car from amazon and assembled it when it arrived. We then looked for a way to send commands to the robot. After reading the documentation we decided to communicate with the robot using an existing TCP socket on the robot, which is also used to communicate with the Robots App.
The commands we send look like this:

{"N": 1 ,"D1": 1 ,"D2": 50, "D3": 1}, {"N": 1 ,"D1": 1 ,"D2": 0, "D3": 1}, {"N": 1 ,"D1": 2 ,"D2": 50,"D3":1}, {"N": 1 ,"D1": 2 ,"D2": 0,"D3":1}

These Commands move one wheel at a time and stop them afterwards. We run these Commands from a Rasberry Pi 4. As we also want to support to control the robot with gestures. We also successfully attached a camera to the Raspberry Pi as well as to get a running live capture.
We also worked on controlling the robot using gestures. For keypoint extraction we use the google media_pipe library. The library can extract 21 keypoint on a hand. We then trained a 3 layer MLP on the keypoints extracted from 3 poses (stop, point left, point right). The model reached up to 97% accuracy while training for less than 100 epochs. In the listing you can see the model used for gesture classification.
We have found a sample EEG dataset that we have downloaded and now want to analyze.
We also tried to use the AVS device SDK. It provides a c++ interface to use Alexa on pretty much any device. Using the sample app we got Alexa to run, so we could use commands like "Alexa tell me a joke". But when we were trying to use the "disco ball" which is like a smart home device, but it always said that those commands can only be used on "certified devices".

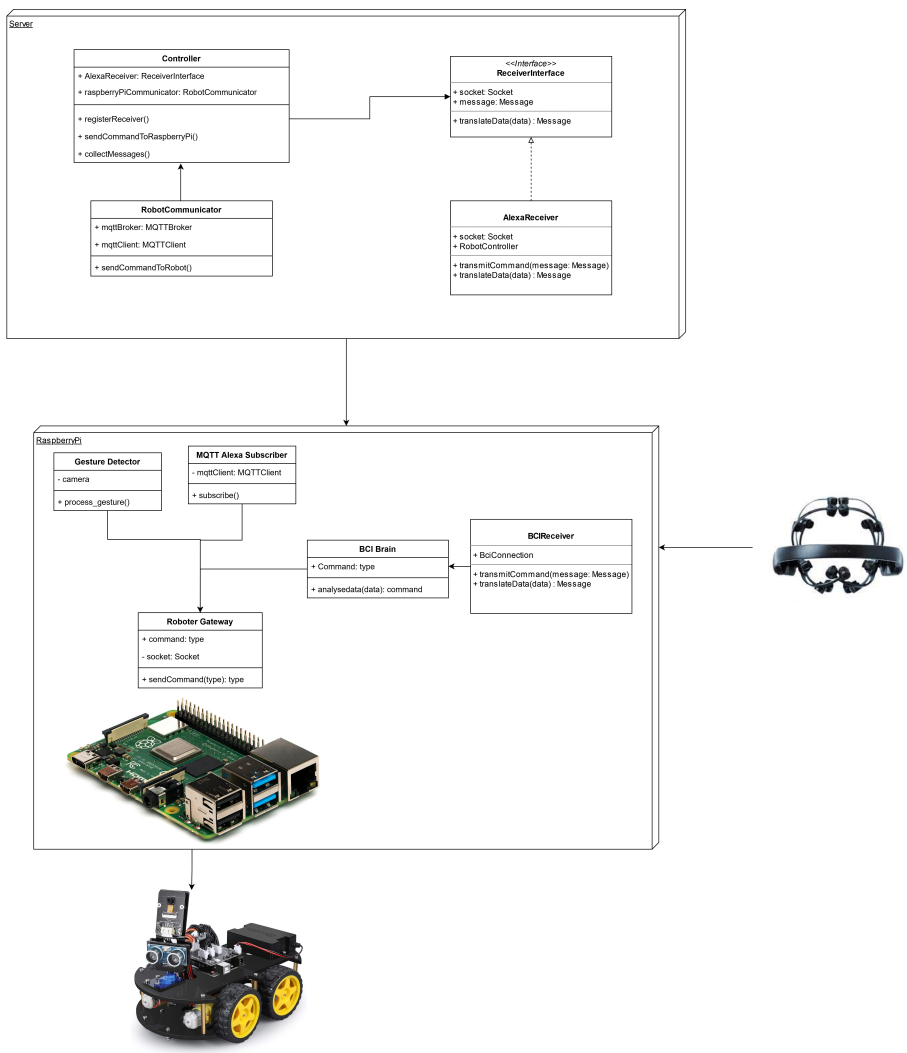
```python
from torch import nn

class LinearReluStack(nn.Module):
    def __init__(self, in_features, out_features, dropout_p=None):
        super().__init__()

        self.linear_stack = nn.Sequential(
            nn.Linear(in_features, out_features),
            nn.ReLU()
        )

        if dropout_p is not None:
            self.linear_stack.append(nn.Dropout(p=dropout_p))

    def forward(self, x):
        return self.linear_stack(x)


class GestureClassifier(nn.Module):
    def __init__(self, n_classes, return_proba=False):
        super().__init__()

        self.linear_stack1 = LinearReluStack(42, 20, .2)
        self.linear_stack2 = LinearReluStack(20, 10, .4)

        self.output = nn.Sequential(
            nn.Linear(in_features=10, out_features=n_classes)
        )

        if return_proba:
            self.output.append(nn.Softmax(dim=0))

    def forward(self, x):
        x = self.linear_stack1(x)
        x = self.linear_stack2(x)
        x = self.output(x)
        return x
```

Architectural Spike MS Report

Team A

December 2022

1 Tech Demo MS



This sprint we worked on the Architecture of our project. After heated debates we decided to try to run as much as possible on the Raspberry Pi 4. Thus the Raspberry Pi should evaluate the EEG Data as well as the Gestures. The Raspberry Pi can not directly process the input of the Alexa thus we will run a MQTT Server which will forward Alexa commands to the Raspberry Pi. The Raspberry Pi then will send all the commands to the robot and mediates between all possible inputs.

2 What did we accomplish in the last Sprint

- we defined the fundamental architecture for our project described in the diagrams above
- further testing and experimenting with technologies to include the new findings in the architectural mode

3 What do we plan for the next Sprint

- implement services for user input:
  - voice commands (alexa)
  - gesture detection
  - brain-computer-interface (machine learning model)
- generate mocking data for the BCI
- implement cloud-server(command prioritizer, routines)
- implement RaspberryPi

4 What hindrances did we have?

- we still have no EEG Headset
- communication from Alexa-Skill to Robot
- uncertainty as where to process the images of the camera
- The RaspbeeryPi has to communicate with the robot and with the internet to get access to the internet
  Thus we plan to connect the robot and the RaspberryPi to a joined WLAN.

Alpha MS Report

Team A

December 2022

1 What did we accomplish in the last Sprint

- implemented RaspberryPi System with command prioritization
- implemented controls via voice commands / Alexa
- implemented controls via hand gestures

2 What do we plan for the next Sprint

- implement brain computer interface receiver and command
- integrate routines / scheduling function to chain commands and execute them
- redefine command prioritization

3 What hindrances did we have?

- EEG headset arrived one week ago, license has been applied
- The RaspberryPi has to communicate with the robot and with the internet to get access to the internet.
  For now we use a mobile hotspot.
- There are not all necessary packages available for the ARM64 RaspberryPi, for example many python 3.10 packages like mediapipe.

Beta MS Report

Team A

January 2023

1 What did we accomplish in the last Sprint

- Added more gestures to the gesture classifier (rock, ok sign, metal sign) to prepare for routines
- Train a model for the eeg headset
- Sending data from the eeg headset to the raspberry pi
- Make the robot drive using the eeg headset
- Adjusted Command-Message pattern to not include the message type (single action / pattern)

2 What do we plan for the next Sprint

- Record gestures for playback
- Execute recorded playback using gestures
- Execute recorded playback using alexa

3 What hindrances did we have?

- Raspberry Pi crashed and deleted the system
- In preparation of pattern recording the code had to be refactored, that introduced some bugs

Implemented Gesture Input

Domenic Rothenberger 78158,
Ali Karami 80018,
Roman Dietenmeier 79709,
Matthias Pijarowski 79673

January 2023

Contents

1 Introduction **3**

2 Gestures registered in our Implementation **3**

1 Introduction

This document lists the different gestures that have been implemented. The other commands for the implemented Receivers utilizing Alexa and the Emotiv Epoc X BCI headset are not further described. You can find the Alexa Skill needed in the src directory. Similar to the Alexa the Emotiv Epoc X EEG headset is dependent on their software to easily train and utilize their headset. With the software available it is recommended to train the headset for every person before every use to guarantee ideal performance.
The implemented gesture classifier on the other hand comes with a trained model fully working with every pair of hands we tested. You can try out the Gesture Classifier yourself after you install the python3 dependencies of the dependencies folders requirements.txt. Then you can run the demo.py file in the src/gestureClassifier/src directory. This will use an attached camera and process the images just as in the following figures.

2 Gestures registered in our Implementation

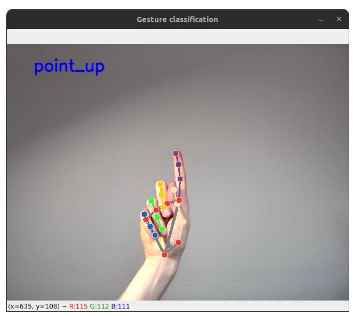


Figure 1: Point Up

The "Point Up" Gesture lets the Robot move forward.

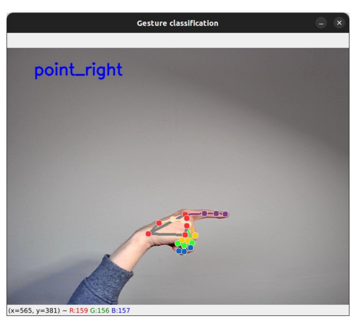


Figure 2: Point Right

The "Point right" Gesture lets the Robot rotate right.

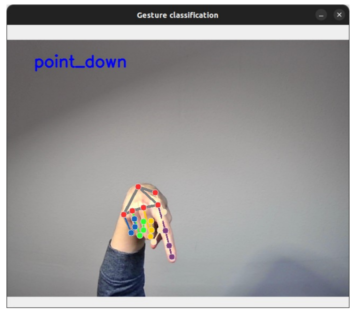


Figure 3: Point Down

The "Point down" Gesture lets the Robot move backwards.

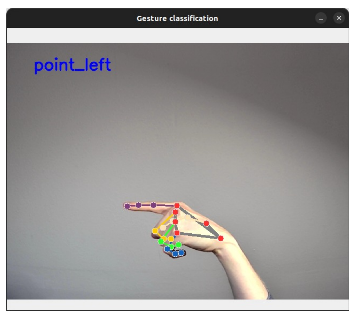


Figure 4: Point Left

The "Point left" Gesture lets the Robot rotate left.

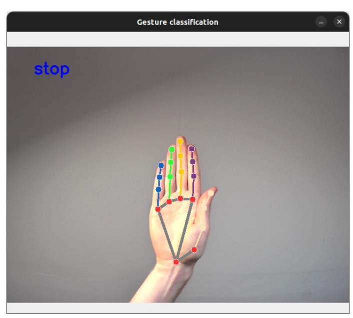


Figure 5: Stop

The "Stop Sign" Gesture starts the Movement / Pattern of the Robot.

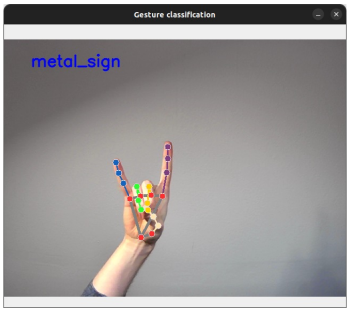


Figure 6: Metal Sign

The "Metal Sign" Gesture starts the Recording of a pattern.

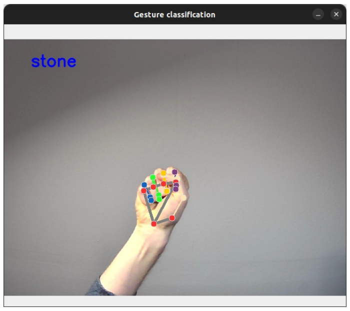


Figure 7: Stone

The "Stone Sign" Gesture represents a Pattern Symbol / "Name".

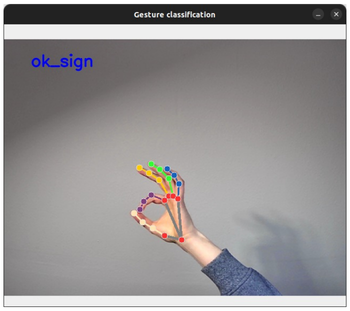


Figure 8: OK Sign

The "Ok Sign" Gesture ends the Recording of a pattern