

Summary

MyoSign is a program that aims to allow the communication between the hearing and deaf communities. American Sign Language, or ASL, is the most used form of communication for the deaf community around the world. With programs like Google Translate and PROMT, it is easy to translate any verbal language to another. ASL, a non-verbal language, uses hand movements to convey a person's thoughts. This poses the tricky problem of how to allow communication between verbal and non-verbal languages. There are very few non-verbal translators on the market but they do not allow mobility and ease-of-use. MyoSign aims to eliminate that. MyoSign uses a gesture device that is worn on the forearm, allowing users to easily use, transport, and calibrate the system.

Users

Individuals who know sign language will use this device to teach the arm band the signs. From these data points, the users will then use the device with a new device listener and start signing, accurately predicting the sign on the screen.

Use Cases

Signing with the device on the arm:

When the user starts the program, they are asked to name the gesture and perform the recording 'x' amount of times. The program will take 13 different data points every millisecond and place this in a csv file. The data points are from the accelerometer, gyroscope, and EMG sensors on the armband. Once the csv file has been created with all the data points, the program will then put this into a directory with all the other gestures.

Predicting the sign:

In this part of the program, the user will place the same armband on their forearm and open up the program. The user will prompt the interface when they will start signing the gesture. Once the user finishes signing the gesture, the program will return the gesture that is the closest to the template modeled by the algorithm.

Major Components & Requirements

Hardware:

Myo Armband

- Bluetooth dongle
- Stainless steel EMG Sensors

- Nine-axis IMU including,
 - o Three-axis accelerometer
 - o Three-axis gyroscope
 - o Three-axis magnetometer
- Dual Indicator LEDs
- ARM Cortex M4 Processor

Software:

Get

Functional Requirements:

- Collect the name of the gesture and the number of times that the user will perform the gesture
- Based off of how many times the user wants to perform the gesture, the program will record the data until they prompt the interface to stop recording
- Output a single csv file with all the data points for an individual gesture (with all records labeled)
-

Non-Functional Requirements:

- After each recording of the gesture, the user has the option to reject or accept the recording
- The program should allow the user to perform the gesture as many times as they would like and be able to record and save the data to a csv file of any size

Analyze/Dynamic Time Warping Algorithm

Functional Requirements:

- Takes in a directory of csv files (the datasets)
- Once the files are properly loaded, the program will input each file into the dynamic time warping algorithm
- Outputs a labelled template for each gesture into a text file labeled DTWMModel
- The algorithm will perform a test based on the datasets input initially
 - o The test will provide a percentage that is the accuracy of the test ran

Non-Functional Requirements:

- The algorithm should perform its analysis within a minute or two of initially running the program
- Once the algorithm runs the test, the percentage should be higher than 50%, if not the user should redo the datasets.

Predict

Functional Requirements:

- The user will prompt the interface to start recording the gesture
- The user will perform the gesture and the program will read/record the data

- Once the recording is done, the program will compare this data to the templates from the algorithm above

Non-Functional Requirements:

- If the gesture is not close enough to any of the templates, then the program will return with a 'null' gesture

Information Flow

