High Level Design (HLD)

Gesture Prediction using Sensors

Revision Number: 1.0

Last date of revision: 01/12/2022

Prakhyath Bhandary

Contents

Document Version Control

Date Issued	Version	Description	Author
01-12-2022	1.0	First Version of Complete HLD	Prakhyath Bhandary

Abstract

Natural control methods based on surface electromyography (sEMG) and pattern recognition are promising for hand prosthetics and to control various devices based on gestures. However, the control robustness offered by scientific research is still not sufficient for many real life applications, and commercial prostheses are capable of offering natural control for only a few movements. In recent years deep learning revolutionized several fields of machine learning, including computer vision and speech recognition.

The goal is to predict the gestures based on the data recorded by the MYO Thalmic bracelet.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) document is to add the necessary detail to the project description to represent a suitable model and coding for application. This document is also intended to help detect contradictions before coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface is implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology stack. The HLD uses non-technical to mildly technical terms which should be understandable to the administrators of the system.

2 General Description

2.1 Problem Statement & Product Perspective

The goal of project is to predict the gesture of user by using sensor outputs from a wearable device. This can be further used for prosthetics or to control gadgets.

2.2 Proposed Solution

For recording patterns, we used a MYO Thalmic bracelet worn on a user’s forearm, and a PC with a Bluetooth receiver. The bracelet is equipped with eight sensors equally spaced around the forearm that simultaneously acquire myographic signals. The signals are sent through a Bluetooth interface to a PC. The signals are processed using machine learning techniques to predict the gesture.

2.3 Technical Requirements

The solution can be a cloud-based or application hosted on an internal server or even be hosted on a local machine. For accessing this applicationbelow are the minimum requirements:

- Good internet connection.
- Web Browser.

For training model, the system requirements are as follows:

- 4+ GB RAM preferred
- Operation System: Windows, Linux, Mac
- Visual Studio Code / Jupyter notebook

2.4 Data Requirements

Data requirements completely depends on our problem statement.

- Signal outputs stored in Cassandra database.

- Input file feature/field names and its sequence should be followed as decided.

2.5 Tools used

- Python programming language.
- Libraries such as Pandas, Numpy, Scikit-Learn, Matplotlib
- Database: MongoDB, Cassandra, S3 bucket
- Framework: FastAPI
- IDE: Jupyter Notebook,VSCode.
- Cloud service: AWS S3, GCP
- CI/CD pipeline : Github Actions

- Git and Github.
- VSCode and Jupyter notebook is used as IDE.
- For Visualization of the plots Matplotlib is used.
- AWS/GCP is used for deployment of the model.
- Front-end development is done using FastAPI.
- Python is used for backend development.
- Github Actions is used for CI/CD pipeline.

- GitHub is used for version control system

2.6 Constraints

MLOPs on the cloud must be fully automated in consideration of continuous integration, continuous deployment with retraining approach of model, and archiving the data over time.

The application should be user friendly as automated as possible. Users can easily use the application and not needed to know any of the workings.

2.7 Assumptions

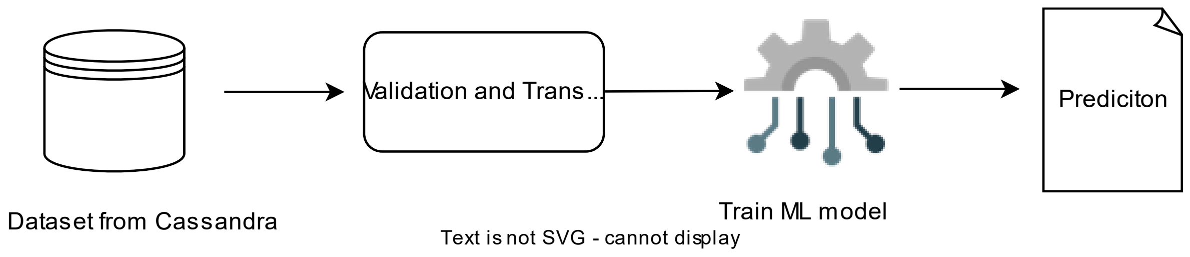
The main objective of the project is to develop an API to predict the gesture based on sensor inputs. Machine learning based classification model is used for predicting above mentioned cases on the input data.

3 Design Details

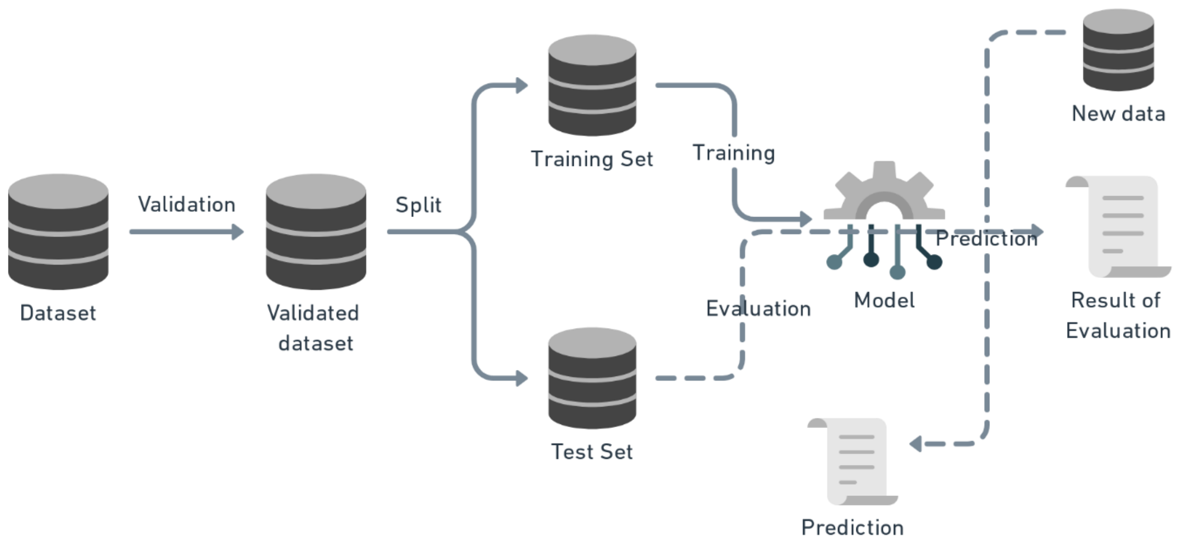
3.1 Process flow

For finding the gesture we will be using machine learning model.
Below is the process flow diagram.

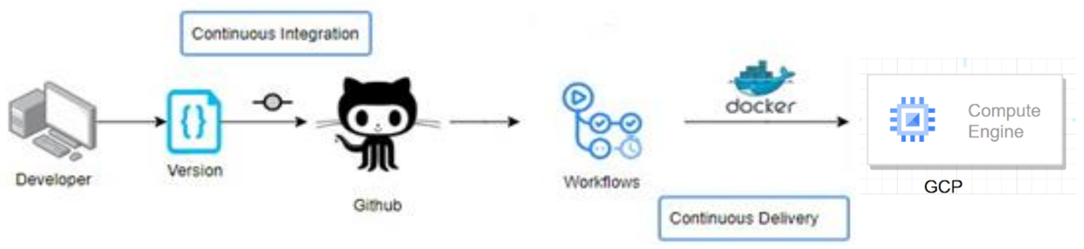
Proposed methodology



3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Event log

The system should log every event so that the track of every detail will be known and what process is running currently could be seen.

Initial Step-By-Step Description:

1. The System identifies at what step logging is required.
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

The system should identify the errors encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.4 Optimization

Data strategy derives performance:

1. Filling missing values.
2. Replacing outliers.
3. Over Sampling
4. Target Encoding.
5. Data standardization.
6. Hyper parameter tuning
7. Validating score again

3.5 Reusability

The entire solution will be done in modular fashion and will be API oriented. So, in the case of the scaling the application, the components are completely reusable.

3.6 Application compatibility

The different components for this project will be using Python as an interface between them. Each component will have its task to perform, and it is the job of Python to ensure the proper transfer of information.

3.7 Deployment

Google Compute Engine

4 Conclusion

In this project, machine learning is used to classify the gestures based on sensor inputs. This can be used in various domains such as prosthetics, home automation etc. Metrics like cross entropy loss is used the measure the accuracy of model.