Imperial College London

Laboratory in Applied Machine Learning

Imperial College London

Department of Electrical and Electronic Engineering

sEMG-Based Silent Speech Interface

Author:
Endong Sun
Jiangnan Ye
Qihan Yang
Wenqiang Lai
Ye Mao

Supervisor:
Prof. Krystian Mikolajczyk
Dr. Ad Spiers

June 26, 2023

Abstract

Regular communication is not always possible. Communication disturbances affect nearly 1 in 10 people, and almost 6 million children have a speech or language disorder. A solution to address the voice disorder problem is recognising speech from articulatory muscles. Hence, surface electromyography-based silent speech interfaces (sEMG-based SSIs) have been proposed and studied for decades. However, most related work is conducted in a strict lab environment and uses expensive sensors. In this project, we proposed an sEMG-based SSI that mounts on Arduino and is feasible for daily usage.

The objectives are assembling the hardware system, collecting data, training machine learning models to classify different words, and evaluating the model performance in a real-time scenario. The hardware includes three MyoWare surface EMG sensors (Sparkfun) and Arduino Nano 33 BLE. Cable shields are used to simplify the data collection procedure. The investigated muscles are the levator anguli oris (LAO: channel 1, Arduino A0 port), the depressor anguli oris (DAO: channel 3, Arduino A0 port), and the mentalis (MEN: channel 2, Arduino A2 port). We follow standard signal processing steps, including denoising with wavelet and feature extraction with sliding window. The models include both traditional machine learning algorithms, such as random forest [1], decision tree [2], and KNN [3], and deep learning models, such as CNN. We further add ensemble techniques to the traditional methods and three-head architecture to the CNN model to increase the accuracy. Finally, we implement Tensorflow Lite to achieve on-board inference.

At this point, solving some data collection and external testing challenges, we successfully fulfilled all the objectives and achieved accuracy up to %86.31. Although the model has good off-line test accuracy, the real-time performance is not satisfying, especially for the unseen subjects. In the following sections, we further explain the methodology and discuss the pros and cons of our system.

Contents

List of Figures

List of Tables

1 Introduction

In this project, we proposed a surface electromyography-based silent speech interface (sEMG-based SSI). In this section, we shed light on the importance of this topic, some history of related work, and the objectives of our study.

1.1 Background

It is vital to know that regular communication is not always possible. According to the service from America Speech-Language-Hearing Association, 40 million Americans have communication disorders, costing the US approximately $154–186 billion annually. Royal College of Speech and Language Therapists reports that up to 14 million people in the UK (20% of the population) will experience communication difficulty at some point in their lives. Many diseases are related to communication difficulties. Aphasia makes patients hard to complacence or formulate language. Apraxia / Dysarthria makes patient hard to plan or program the articulator movements. Voice disorders (dysphonia) cause disturbance in the vocal folds or other organs involved in voice production (e.g., laryngeal cancer).

Aphasia and apraxia are brain injuries that require neuron activity sensors to intervene. However, voice disorder can be addressed by recognizing speech from speech-related muscle signals. Hence, sEMG-based SSIs have been proposed. The use of sEMG for speech recognition dates back to the mid-1980s; however, competitive performance was first reported by in [8]. In 2002, the Japanese company NTT DoCoMo announced it had created a silent mobile phone using sEMG and imaging of lip movement [9]. From then on, much research about sEMG-based SSIs has been conducted.

1.2 Scope and Objectives

Though much research has been done on silent speech, the problems remain. Most experiments are conducted in a strict lab environment and use expensive sensors. For example, Trigno Mini sensors (Delsys, Inc, Natick, USA), specifically designed to provide a miniaturized sensor interface that better conforms to the face and neck and mitigates sources of movement artifact and sweat-build-up during the relatively longer experiments, are used in [10]. In [11], high dense EMG electrode arrays, acquired from OT Bioelettronica, are used to cover the major articulatory muscles and get more robust data.

However, those experiment setups may not be suitable in practice. Hence, in this project, we proposed a sEMG silent speech interface that mounts on Arduino and is feasible for daily usage. The objectives of this work include:

- Use affordable sEMG sensors and Arduino to ensemble a silent speech interface that can classify a few words.
- Use the ensembled hardware to collect and construct a dataset.
- Apply signal processing to the sEMG signals and extract usable features for the

traditional machine learning model.

- Train several machine learning or deep learning models to classify the data and evaluate the performance.
- Achieve real-time on-board inference.

1.3 Achievements

At this point, we successfully fulfilled five objectives except for the last one. In the following sections of this report, we will explain in detail how we implement each component, the difficulties we met, how we address the issue, and what remains to be done.

1.4 Overview of Dissertation

Section 2 explains the existing literature. Section 3 demonstrates our hardware design. Section 4 illustrates the data collection protocol. Section 5 contains the methodology we applied. Section 6 provides the result and the main discussion based on that. Section 8 concludes the whole project and lists some future work.

2 Literature Review

2.1 Silent Speech Interface

Silent speech interface (SSI) is a range of products designed for speech communication of the voice disorders who are not able to generate regular audio signals. Sensors in SSI aim to capture non-acoustic bio-signals which should vary in different activities.

As shown in Figure 1, the biosignals used to monitor the speech are mainly from two classes, brain activities and muscle activities. In terms of brain activities, haemodynamic response sensors can be used to detect the changes in blood oxygenation, which reflects the stimulus in neural activity during cognitive tasks. In addition, the electric currents generated by the extracellular medium during neuronal activity can also be captured to analyse speech patterns[4]. The spikes caused by the synaptic transmembrane current are the major contribution[12]. The components of these currents give rise to the electrical fields which electrodes can record. These recordings are of high time resolution and can be interpreted reliably[13]. Notably, an alternative non-invasive method for detecting such electrical fields is the well-known Electroencephalography. However, the detection of brain activity signals requires high accuracy equipment that is costly and beyond our project's budget.

Therefore, monitoring muscle activities becomes an alternative method. As mentioned, the brain makes the decision of speech, which will then activate muscles to produce voice by sending electrical signals that stimulate the facial muscles to relax and contract. These electrical signals can be measured with electromyography(EMG) which records the electrical potentials introduced by the depolarisation of the external membrane of the muscle fibres[14]. The EMG can be measured with two types of electrodes: invasive and

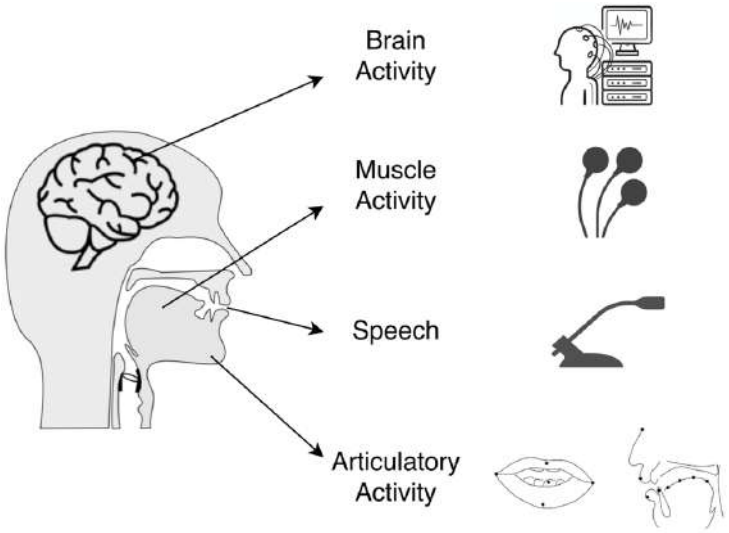


Figure 1: Silent speech based on brain activity method and non-invasive EMG method [4]

non-invasive electrodes. The invasive electrodes insert needles into the muscle, which is accurate for measuring local parts of muscles. The non-invasive electrodes are directly attached to the skin's surface, which is preferred by industrial because it is safe and more sanitary. The surface EMG (sEMG) is chosen as the method for this project.

The sEMG signal measuring the stimulus from the brain to muscles has been proved to be effective in retrieving the speech signal. Moreover, this feature still holds even for inaudible speech, which means that the voice is not generated[15]. Therefore, the relationship between sEMG signals and speech can be modeled to help persons with voice disorders communicate. It is worth mentioning that the occurrence of EMG signal can usually be $60ms$ ahead of articulatory motion, which is beneficial in reducing the latency of silence speech[16].

Works related to the sEMG method have made prominent progress in recent years. An array-based method is introduced by [11] which aims to solve the problem caused by the high dimension of EMG signals. This work presents a system with 10.9% error rate on audible speeches of 108 words. [17] extracts articulatory features using hamming-windowed Short Time Fourier Transform(STFT), which can be further utilized for silent speech predictions to replace the classical EMG features. The F-score of the articulatory feature classifiers was improved from 0.467 to 0.502. A domain-adversarial training method on neural networks is proposed in [18] to reduce the effect of reattaching EMG electrodes on the recognition accuracy. The relative Word Error Rate is reduced by 17.0% on the development dataset and 4.7% on the evaluation dataset. For a similar problem, [19] applies transfer learning to CNN and SVM, which can be used to calibrate the shift of the sEMG electrode within 2.5 cm. This problem can also be alleviated with suitable normalization and adaptation methods that are session independent, as introduced in [20], bringing recognition rates up to 87.1%. [21] provides an insight into the effect of channel number on the sEMG silent speech recognition. It finds that a smaller channel number plus another few electrodes for recording detailed information can be the most

computationally-effective and time-efficient choice, where the results of classification accuracy are up to 80%. In the language of Portuguese, [22] makes a successful attempt and provides a potential state-of-the-art accuracy, in the best-run case, reaching 70%. They also explore the challenge of nasality in classification for Portuguese. [23] breaks the barrier of separate words prediction and extends sEMG silent speech to a continuous domain with carefully selected features.

2.2 Signal Processing

2.2.1 Noises in Bio-signals

There are many external and internal noises in the bio-signal, especially when facial muscles are involved. Due to the existence of electronic devices and physiological factors, various background noise will be generated during the data collection. Noises are coming from [24]:

1. Electrode noise: In general, the larger the electrode size, the smaller the impedance. The limited size of the electrode makes it difficult to reduce the impedance, which further reduces the signal quality and gives a low signal-to-noise ratio (SNR).

2. Cross-talk: the facial muscles are dense, and the activities of the undetected muscles could affect the desired signals.

3. Internal noise: The number of muscle fibers per unit, depth and location of active fibers, and amount of tissue will inevitably affect data quality.

4. Static noise: Electrical equipment in the experimental environment will generate static noise.

5. Irreducible noise: In addition to the noise listed above, what affects the sEMG signal most is the ECG artifact and the electromagnetic interference at 50 and 150 Hz [25, 26]. This kind of noise can be limited with signal post-processing algorithms.

2.2.2 Denoising

Wavelet has been growing in popularity as an alternative to the usual frequency domain filtering method. The main problem for fast Fourier transform (FFT) and short-term Fourier transform(SFT) is that they can only treat the signal as stationary discrete points. However, continuous wavelet transform can approximate the continuous signal by integrating it at different resolution levels [27]. This method has been verified as an efficient denoising method for biosignals [28]. For sEMG signal, wavelet achieves good results in removing the interference of random noises(e.g., white Gaussian Noise(WGN))[29].

Wavelet transform is similar to Principle Component Analysis (PCA) to some extent, which projects the original signal with the wavelet transform to the orthogonal space[30]. A number of threshold strategies perform key roles in denoising. The coefficients of the projected signal are selected to maximally maintain the characteristics of the signal while eliminating noisy components[7]. After isolating the noise in the wavelet domain, the

signal is reconstructed, and the most important signal components will be preserved[31].

The parameters required for the wavelet denoising algorithm can be concluded into five: the wavelet basis function, the scale, the threshold selection rule, the threshold rescaling method, and the thresholding function.

Among these parameters, the selection of the wavelet basis function is the most important factor, which depends on the scenario and characteristics of the signal [24]. The functions should have continuous derivatives, making it convenient to decompose the continuous signals into different levels of components. Among the basis functions, Daubechies's function is long-length orthogonal filters with better energy concentration than the short-length ones [32]. Other common wavelet basis functions are Haar, Meyer, Mexican Hat, Gaussian, Shannon, Biorthogonal, and Morlet[24].

2.3 Machine Learning

Traditional machine learning models are wildly used in bio-signal classification problems. Feature extraction reduces the dimension and extracts useful information from the original data. Data augmentation synthesizes new data to increase the input hypothesis space.

2.3.1 Feature Extraction

Three categories of features can be extracted and analysed for sEMG signal according to [33], which are time-domain, frequency-domain, and time-frequency domain features. Hudgins et al. proposed the mean absolute value (MAV), mean absolute value slope, slope sign changes (SSC), waveform lengths (WL), and zero crossings (ZC) in [8]. Time-domain features are verified as the optimal solution for feature extraction of EMG signals in [34]. Frequency-domain features are extracted by classical power spectral density estimation. In [35], they proposed the Auto-Regressive coefficients (AR), Modified Frequency Median (MFMD), and Modified Mean Frequency (MFMN) method for analysis of EMG pattern recognition. Median frequency (MDF), Mean frequency (MNF), and Peak frequency (PKF) are extra algorithms for extracting frequency domain features [36]. Frequency ratio (FR), Mean power (MNP), Total power (TTP), and Maximum power (MP) is also popular algorithms for EMG signals to extract frequency-domain feature [37]. The time-frequency domain features, effective feature sets especially for transient myoelectric signal pattern classification, can be extracted using SIFT[17][38] etc. Wavelet Transform (WT) and Wavelet Packet Transform (WPT) were also employed for time-frequency domain features extraction [33].

2.3.2 Data Augmentation

Data augmentation serves for the machine learning models. It is a method of generating synthetic data, expected to cover the unexplored input space[5], to improve the generalization ability of trained models.

The task is focused on dealing with time-series data augmentation for sEMG signals. As shown in Figure 2, for the time domain, direct methods include injecting noise and flipping. Window cropping is another method that has been widely used in the computer

vision realm. It randomly extracts slices from the original signals while assigning them to the same labels. Besides, Window warping, which is similar to dynamic time warping, compresses or extends signals from a random time range. It changes the total length of the signal and, therefore, should be combined with the window cropping method[5]. This project does not include augmentation methods from other domains or more advanced methods.

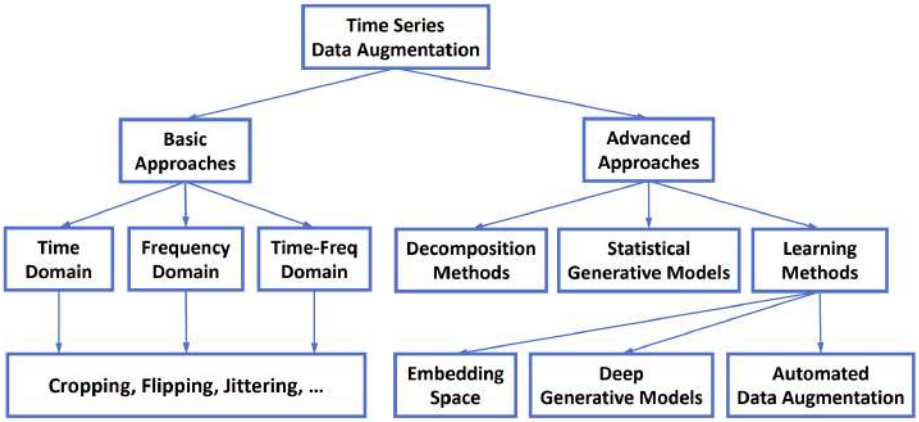


Figure 2: Taxonomy of time series data augmentation techniques [5]

2.3.3 Models and Networks

A number of machine learning techniques have been successfully applied in silent speech tasks. In terms of classical machine learning, Gaussian Mixture Models are a series of techniques commonly seen in voice conversion tasks[39], which perform a weighted integration on different gaussian distributions found with maximum likelihood methods. This method is adapted to a GMM-based-articulatory-to-acoustic mapping model introduced in [40]. [18] also uses GMM as the frontend for its model. The GMM is followed by the HMM backend and the training process is designed to be adversarial, where two networks are optimized simultaneously. Support Vector machine(SVM) is used to classify vowels in [41], where SVM outperforms other machine learning methods e.g. KNN, Bagged Trees and Naive Bayes Kernel. To reduce the channels, thus the number of electrodes attached to the face of subjects, [42] proposes a channel reduction model based on Decision Tree and obtains a word accuracy of 95.17% under merely 5 channels.

Deep Learning, which is featured with end-to-end prediction, also constantly involves in silent speech. Convolutional Neural Network (CNN) is applied in [19] combined with transfer learning techniques to resolve the error caused by repositioning the electrodes. In [43], the EMG signal is directly transformed into speech signal using deep neural network after feature extraction. Moreover, mentioning the EMG-to-speech mapping, Long-Short-Term-Memory (LSTM) cannot be ignored. [44] uses a bidirectional LSTM network with two hidden layers consisting of 100 and 80 memory blocks to directly convert EMG signal to speech.

3 Hardware Design and Implementation

The hardware involved in this project includes surface EMG sensors, Arduino, and additional components such as stripboard, socket, and acrylic cases (Figure 3). The following sections explain the motive behind each choice, provide insights into the building process, and discuss the challenges we faced and how we address the problem. (Five protocols are shown in 19)

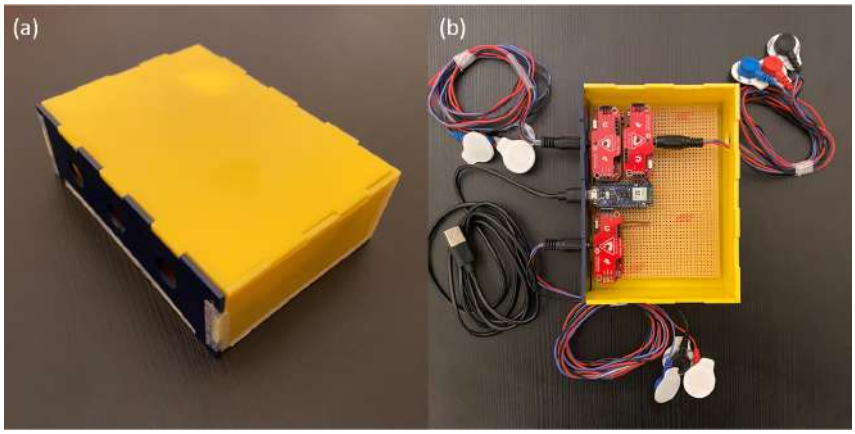


Figure 3: The hardware built for this project: (a) demonstrate the self-contain box; (b) shows all the components involved.

3.1 EMG Sensors

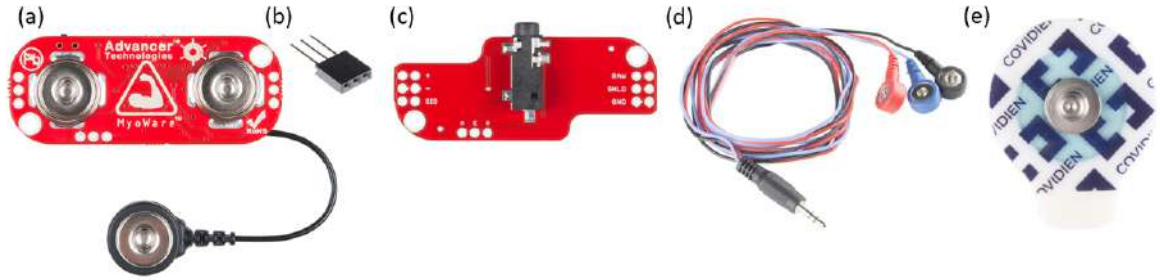


Figure 4: The EMG sensor ensemble: (a) MyoWare EMG sensor; (b) 3-pin header; (c) Cable shield; (d) Sensor cable; (e) Electrode

The sensor used in this project is MyoWare Muscle Sensors (Figure 4(a)). The reason for choosing this sensor lies in two aspects. On the one hand, this sensor is Arduino-powered at an affordable price, which matches the objectives and requirements of our project. On the other hand, the MyoWare board functions by measuring the filtered and rectified electrical activity of a muscle, outputting 0-Vs Volts depending on the amount of activity in the selected muscle, where Vs signifies the voltage of the power source. The rectification simplifies our data preprocessing procedure.

Since the muscle groups on the face are small, attaching the sensors directly to faces is impossible. Hence, we introduce the cable shield (Figure 4(c)), which provides a 3.5mm

jack, allowing us to attach an additional three-electrode sensor cable (Figure 4(d)). The cable shield and sensor are connected with 3-pin headers (Figure 4(b)).

Finally, the sensor cable is attached to the subjects' faces through electrodes. Considering the budget limit, we select the pre-gelled Ag/AgCl surface electrodes with a reasonable diameter ($24mm$) to attach to faces (Figure 4(e)). Ag/AgCl electrodes also generate low electrode-skin interface impedance and low noise levels during biological signals recording [45].

3.2 Arduino

Figure 5: The Arduino Nano 33 BLE

The Arduino selected is Nano 33 BLE (Figure 5). It is an Arduino's $3.3V$ compatible board in the smallest available form with a size of only $45 \times 18mm$. This feature makes our hardware more flexible. Arduino mainly acts as a system's processer that controls the data flow, the sample rate is 1000Hz. Moreover, Nano 33 BLE has a 32-bit ARM Cortex-M4 CPU running at 64 MHz, allowing us to make onboard inferences with the help of support software such as CMSIS-NN library, which will be introduced in the following section of this report.

3.3 Additional Components

We add several extra pieces of hardware to make the whole system self-contained, including stripboard, socket, and acrylic cases.

3.3.1 Stripboard

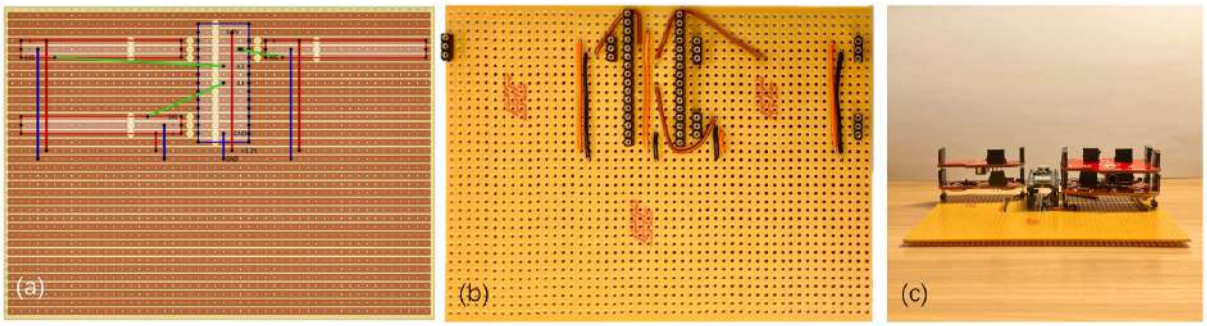


Figure 6: (a) Stripboard circuit design; (b) Stripboard; (c) After attaching the sensors to the stripboard

Stripboard is essentially a prototyping board that allows us to assemble three sensors and the Arduino together without jumping wires. The circuit design is shown in Figure 6(a), where the power supply is displayed in red wire, the GND end is blue, and the EMG signal is green. The actual board after soldering is shown in Figure 6(b).

3.3.2 Socket and Acrylic Cases

Figure 7: Sockets

As shown in Figure 6(b)(c), the sensors and Arduino are not directly soldered on the stripboard but through sockets (Figure 7). The reason behind this is the sensor is very sensitive to the environment. If the sensors touch the stripboard, the static surface electricity will affect their readings and bias the collected signals as discussed in section 2.2.1. Sockets not only allow us to plug in and out the components easily to debug the system, but also provide a distance between the sensors and the stripboard surface that prevent environmental noise.

To add further protection to our system, we laser-cut an acrylic case. On the one hand, it stores all the system's essential components, such as the cables and electrodes. On the other hand, it provides a cleaner environment for the sensors. Furthermore, we add a slot on the bottom of the case, making it suitable for attaching to an armband.

4 Data Collection

In this section, we illustrate the data collection procedure. Details include the muscle selection, the user interface, and the collection protocol. At the end of the session, we provide some data examples and shed light on the general properties of the data.

4.1 Muscle Selection

Facial muscles are a group of about 20 flat skeletal muscles lying underneath the skin of the face and scalp (Figure 8). Compared with the muscle group in the upper limb, facial muscles are more complex and hard to isolate. It is crucial to identify and select the muscles related to human speech, and the location of the electrodes should avoid the overlap area between different muscles to reduce cross-talk.

Many research explores collecting EMG signals from facial muscles. The most commonly used are the levator anguli oris [8, 38, 46, 42], the zygomaticus major [8, 38, 46, 42], the platysma [8, 38], the depressor anguli oris [8, 38, 46, 42], the anterior belly of the digastric [8, 38, 42], and the mentalis [47, 48, 42].

After evaluating the upper muscles with our sensors, we figured out that the zygomaticus major and the digastric suffered severe saturation problems during data collection, and

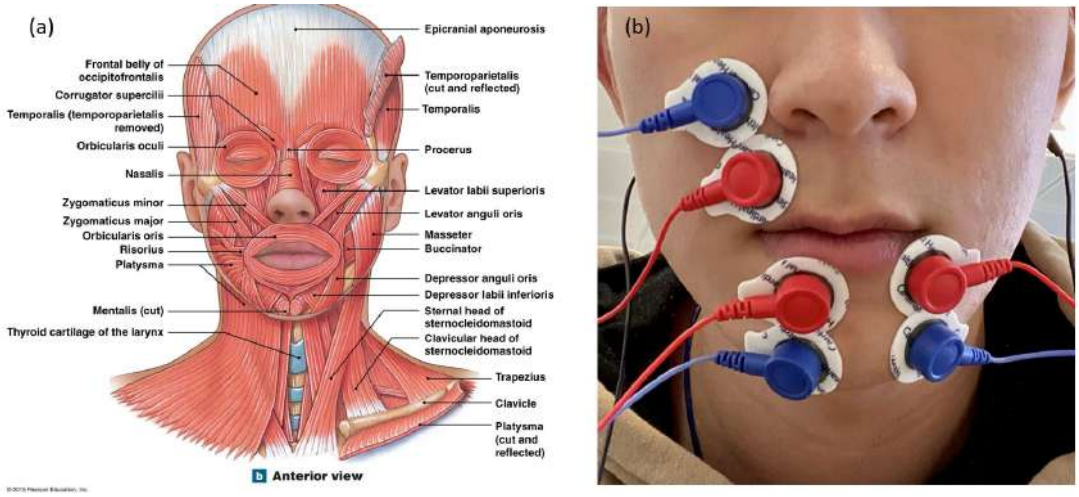


Figure 8: Sensor placement: (a) The anatomy of facial muscles, the photo is adapted from [6]; (b) A demonstration of the sensor placement of this project

platysma, being a thin sheet-like muscle, is hard to locate. Thus, the final selected muscles are the levator anguli oris (LAO: channel 1, Arduino A0 port), the depressor anguli oris (DAO: channel 3, Arduino A0 port), and the mentalis (MEN: channel 2, Arduino A2 port) (Figure 8(b)). To minimize the cross-talk between muscles as discussed in section 2.2.1, we keep the distance between the electrodes around $1cm$.

4.2 User Interface and Collection Protocol

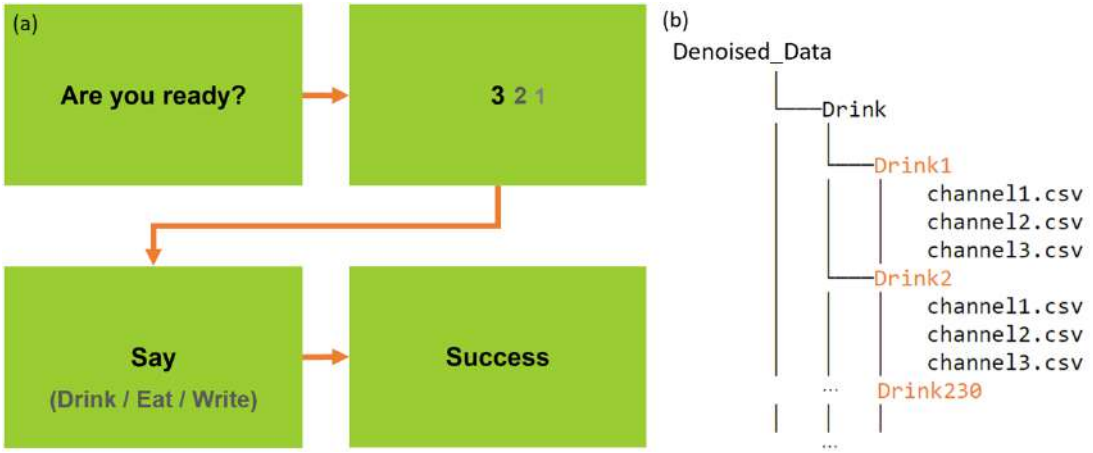


Figure 9: (a) User interface internal logic; (b) data folder structure

To normalize the data collection procedure, we introduced a simple user interface. As shown in Figure 9, the UI first asks whether the subject is ready or not, and the subject can press Enter key to start the data collection. After the confirmation, a target word will appear on the screen after a count down from three. The subject has three seconds to pronounce the word (one from "Drink," "Eat," "Write," and "Say") without voice, resulting in 3000 sample points. If the communication between PC and Arduino works well, the word "Success" will appear on the screen after three seconds. Then the subject can press Enter key to start collecting the next word.

In each data collection session, one subject is asked to follow the UI and repeat words one by one. He/she will take a short break after repeating the word ten times. In this project, we conducted five data collection sessions and recorded data from five subjects in our group. Four of us collect 40 trials for each word, and the remaining one collects 50 trials for each word. Hence, we have 210 trials per word and 840 data in total.

The UI and the collection protocol not only helps the subjects concentrate and save collection time, but also automatically constructs the dataset in a systematic way. After finishing the collection, all the data are stored in a folder as shown in Figure 9.

4.3 Data Preliminary Analysis

4.3.1 Artifact and Saturation

Two significant difficulties in data collection are artifacts and saturation problems. Artifacts are mainly introduced by sensors' sensitivity and cable connection. As introduced in previous sections, we address these issues by adding sockets and protective cases to create a stable environment for sensors. The saturation problem, on the one hand, is solved by wisely choosing proper muscles that provide evident but not overlarge signals. On the other hand, we manually delete the saturated data sample during data collection, which causes extra time but guarantees the data's usability.

4.3.2 Data Visualization

Figure 10 display a grid of data shows different words from different subjects, in which LAO is displayed in blue, MEN is in orange, and DAO is in green. We see different subjects have different word patterns. Combining data from different people to train one model significantly increases the training difficulty, but improves the model's robustness.

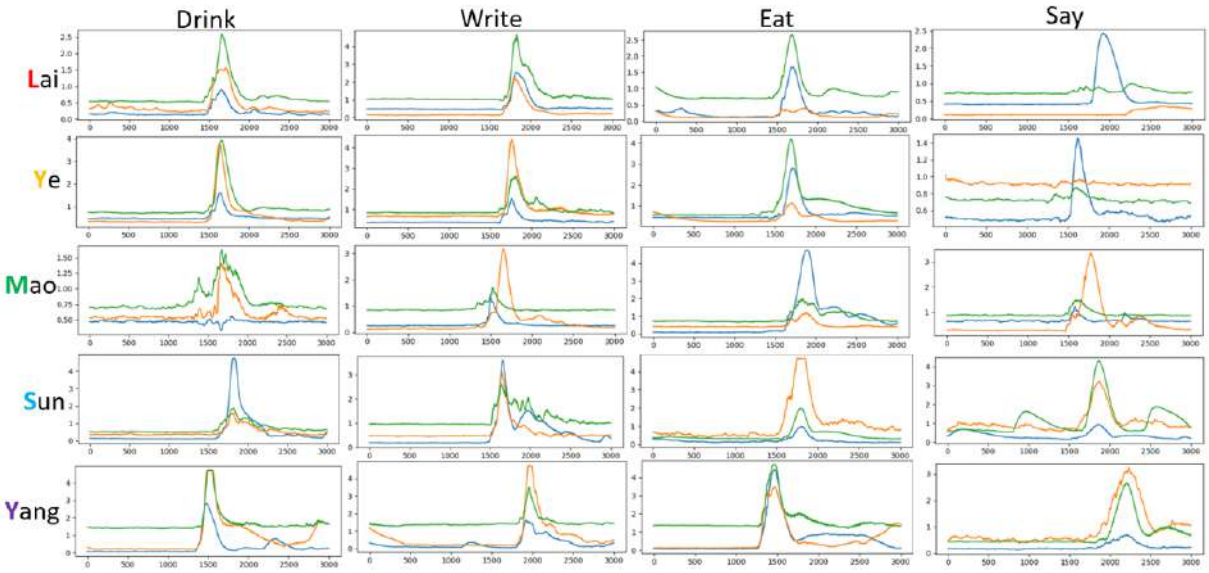


Figure 10: Data visualization.

5 Methodology

5.1 System Design

We follow the standard EMG signal processing steps, including signal processing, feature extraction, model training, and result evaluation. The overall system design is shown in Figure 11.

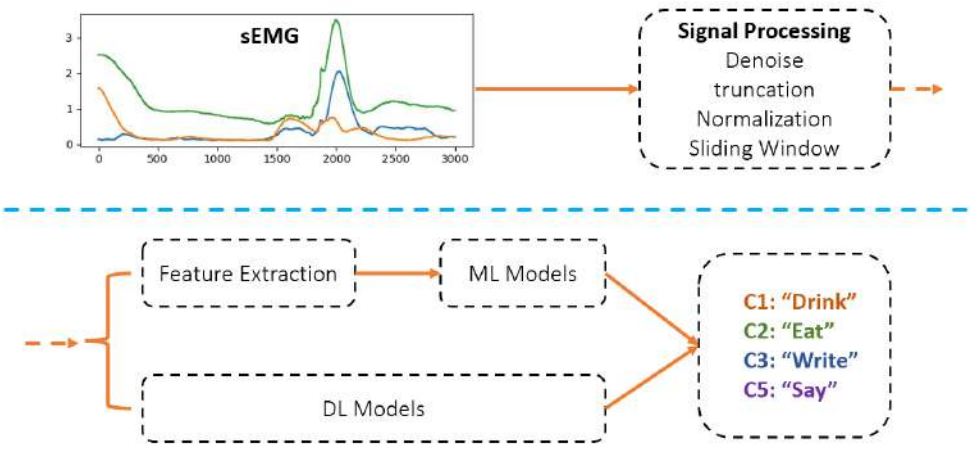


Figure 11: System Design

5.2 Signal Processing

In this part, we perform denoising of the original data. A part of denoising has been done physically with hardware design components, such as using an acrylic box and sockets on the stripboard as insulators (section 3.3.2). The following signal post-processing is listed in the following sections.

5.2.1 Transient State Elimination

The sEMG signals are generated from the summation of several motor unit action potential trains. Signals are non-stationary, non-linear, and stochastic. Two states are defined for sEMG signals: the transient-state and the steady-state. The transient-state in sEMG signals is described as the bursts of myoelectric activity that accompany sudden muscular effort while executing the movement. This state can be observed sometimes in the first 1000 sample points, generating undesired noisy data (Figure 11). When the muscular effort is in a sustained contraction to reach the muscle's final position, and the muscle length is no longer modified, the steady-state is achieved[49]. sEMG signals are regular and significant in the steady-state, which is the sample points from 1000 to 3000. Therefore, we cut off the first 1000 sample, guarantee the remaining samples are in the steady-state.

5.2.2 Denoising

Figure 12 illustrates the flowchart of the wavelet denoising process applied in our project (section 2.2.2). In the wavelet decomposition section, we use Daubechies filter of order 4

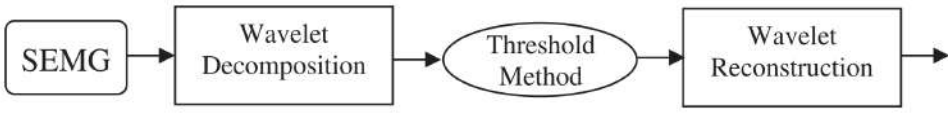


Figure 12: The general procedure of wavelet denoising [7]

(DB4) as the basis function. The signals are projected onto the DB4 basis functions (by convolution with the filter) in a hierarchical manner (Figure 13). The outputs of the later transform indicate more refined components that belong to high-frequency parts. The projected coefficients on a different basis form a new subspace called the wavelet space.

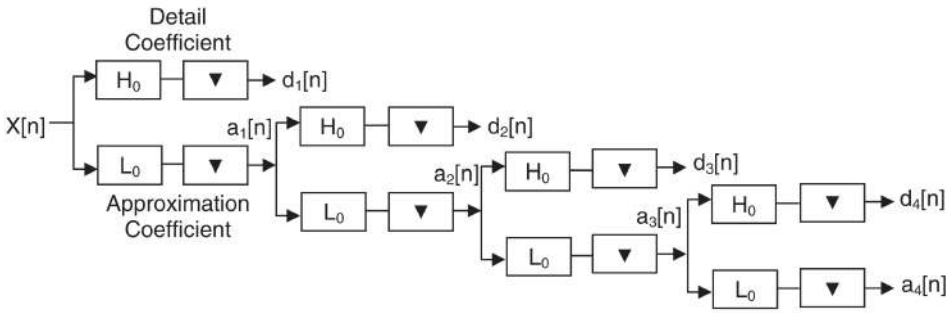


Figure 13: Wavelet transform with DB4 filter [7]

Noises are commonly contained in the high-frequency components; therefore, the denoising process is done by thresholding these coefficients. Coefficients with a larger magnitude contain more important information and less instead. Thus, smaller coefficients can be considered as noises and be eliminated. In this project, 'soft' thresholding is performed. The threshold is calculated according to VisuShrink, which is a prevalently used fast and easy threshold method in wavelet[50]. The formula is shown in Equation 1, where σ is the standard deviation of noise and n is the number of samples.

$$\lambda = \sigma\sqrt{2\log(n)} \tag{1}$$

The soft thresholding follows Equation 2 where a smooth transition is present when smaller values are eliminated.

$$y_{soft}(t) = \begin{cases} sgn(x(t)) \cdot (|x(t) - \delta|), & |x(t)| > \delta \\ 0, & |x(t)| < \delta \end{cases} \tag{2}$$

Finally, the thresholded components are reconstructed from wavelet space into time-space, and the reconstruction results are the denoised signals.

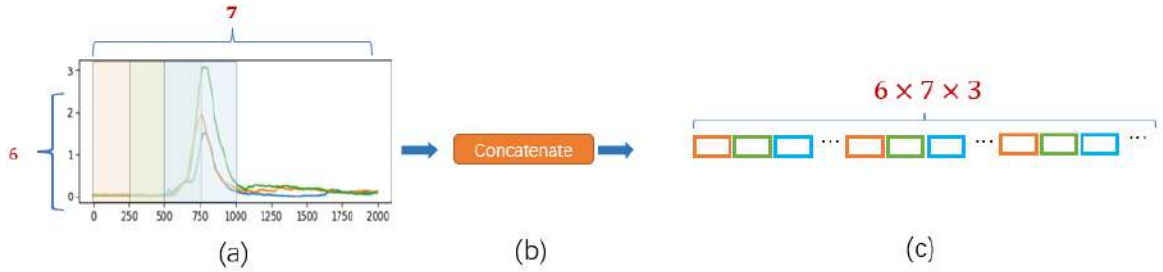


Figure 14: Feature extraction: (a) 6 feature extraction methods on one signal with 3 channels; (b) Concatenates features across all 3 channels; (c) New data with 126 features

5.2.3 Feature Extraction

Each data consists of three channels corresponding to three muscles locations, with 2000 data points per channel after eliminating the transient state. A sliding window of size 500 with step 250 is applied to the sEMG signals. Hence, we obtain $((2000-500)/250+1)\times 3 = 21$ windows per data. From each window, we used six features in the frequency-domain, including Frequency ratio (FR), Mean power (MNP), Total power (TTP), Mean frequency (MNF), Median frequency (MDF), and Peak frequency (PKF). Features captured from each window from three channels are concatenated to establish the feature vector for the traditional machine learning model. The total number of features per data are: $21 \times 6 = 126$, as shown in Figure 14.

5.2.4 Time Series Data Augmentation

The data augmentation method should be considered according to the specific tasks. In our project, since noise is inevitably present during data collection of sEMG signal, the first augmentation technique is to add White Gaussian Noise (WGN)[51] (section 2.3.2). Moreover, inspired by the IMU sensors used in [52], it is mild to assume that the sEMG sensors can output the detection results at a slightly different speed. Therefore, window-warping is applied to the signal as the second augmentation technique. A random time range of the signal is selected, and the speed of timeline is changed within this time range of signal[53]. The two techniques are then randomly performed on each signal. After augmentation, the amount of samples in the dataset is increased from 840 to 2520.

5.3 Traditional Machine Learning

5.3.1 Single Machine Learning Classifier

As illustrated in section 5.2.3, 126 frequency-domain features are used in a machine learning-based technique. Afterward, the obtained features will be preprocessed with min-max normalisation and PCA analysis to establish a new dataset. The reconstructed dataset will be the input of a number of traditional machine learning classifiers, including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree, Gaussian Naive Bayes, Random forest, and Multi-layer Perceptron (MLP). The random forest classifier delivers the best performance among all models based on the concrete experimental analysis.

5.3.2 Stacking Machine Learning Classifier

Since sEMG signals obtained from different samples at different times are significantly varied, the expressive power of the single ML classifier is too weak to handle data with certain variability. We thus proposed a novel stacking machine learning algorithm, focusing on effectively merging the prediction results from multiple well-performing base-level ML classifiers trained on the augmented dataset. To start with, our approach, like the traditional ML algorithm, necessitates an explicit feature engineering procedure. After ranking the performance of all classical ML classifiers on the reconstructed dataset, we selected Random Forest, Decision Tree, and K-Nearest Neighbor algorithms as our base-level classifiers. Furthermore, because the number of features compared to the size of the augmented dataset is tiny, PCA analysis is disabled in our stacking approach to preserve as much information as possible in the data.

As a meta-level classifier, a logistic regressor will finally be utilised to learn how to efficiently stack all of the classification results from the base estimators. To generate a training set for training the classifier, a 10-fold cross-validation procedure is applied.

5.4 Deep Learning Methods

5.4.1 Naive Convolutional Neural Network (CNN)

Deep learning methods could be considered as end-to-end systems due to their ability of automatically extracting features from the raw data. Thus, deep learning methods may extract and learn better representations than classic methods. As a starting point, we propose two naive CNN architectures (Table 1). ReLU is the activation function in all layers, except for the output layer, where Softmax is used. The naive models mainly served to examine whether the dataset is learnable for deep learning methods.

Naive 1D-CNN	Naive 2D-CNN
Conv1D(16,k=60,s=30)	Conv2D(8,k=(60,3),s=(30,1))
MaxPool1D(p=6,s=3)	MaxPool2D(p=6, 3,s=(3, 1)
Flatten()	Flatten()
Dropout(0.4)	Dropout(0.4)
Dense(32)	Dense(32)
Dense(4)	Dense(4)

Table 1: Model summaries of Naive 1D-CNN and Naive 2D-CNN

5.4.2 1D Three-head Convolutional Neural Network

To further release the potential abilities of deep learning, we modify a novel approach, referred to as three-head CNN, to perform silent speech recognition. Three-head CNN also allows end-to-end classification without an explicit feature engineering process. Furthermore, our model can process each sEMG signal with a different sized kernel, extracting more meaningful features at various resolutions.

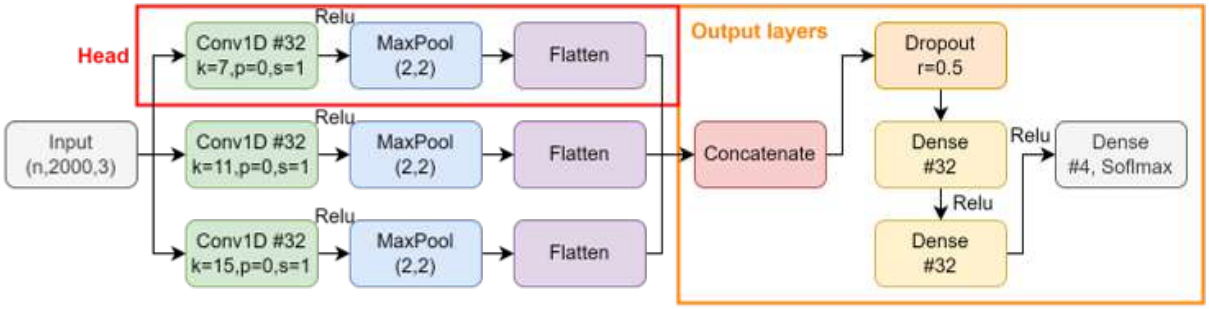


Figure 15: 1D Three-head Convolution

An overview of the three-head CNN architecture is presented in Figure 15. It clearly shows that each convolution head is comprised of three components: a 1D convolutional layer, a 1D Max Pooling layer, and a dropout layer to filter half of the training parameters randomly. The input size for each head is $H \times D$, where H is the number of timesteps of each signal (i.e., 2000) and D represents the number of channels. The reason for using 1D convolution and pooling operation is that the sEMG sensor data used in this work is time-series data. Furthermore, due to the length of the sEMG signal, the convolutional kernel sizes of the three heads are bigger than standard CNN kernel sizes, which are 7, 11, and 15, respectively. In addition, `ReLu` activation and `He Normal` kernel weight initialisation is applied throughout the architecture to speed up the model convergence. Also, *L1* regularisation is included to reduce model complexity by penalising $\|\omega\|$, where ω is a set of neuron weights. Since each non-zero weight contributes to the penalty, it forces weak features to have zero weight. Hence, *L1* regularisation also produces sparse solutions, inherently performing better feature selection from long sequences of inputs like ours. All of the techniques presented thus far are intended to enable each CNN head to build feature maps in a timely and robust manner.

The second component of Three-head CNN is a feed-forward neural network that takes as input the concatenated feature maps from the three CNN heads. The feed-forward neural network is composed of two fully-connected layers, each with 32 units. The output layer comprises four neurons corresponding to the four classified words: "Drink," "Eat," "Write," and "say". Because we are attempting to solve a multi-classification problem, the loss function used is categorical cross-entropy.

5.4.3 2D Three-head Convolutional Neural Network

When we "speak" the word, our three muscles contract simultaneously, indicating implicit correlations between different muscles' EMG signals. Therefore, we propose a Three-head CNN based on 2D convolutions to capture such correlations.

As illustrated in Figure 16, we replace all 1D convolutions in each CNN head with 2D convolutional operators. Since 2D input size is necessary for 2D convolutional layers, we expand the dimension of the original data by one. Similar to the CNN applied to 2D natural images, the width of the image in our work is the number of time steps, and the height of the image represents the number of muscles (i.e., the number of channels in the original Three-head CNN). However, solely three muscles are analysed in our study, so we

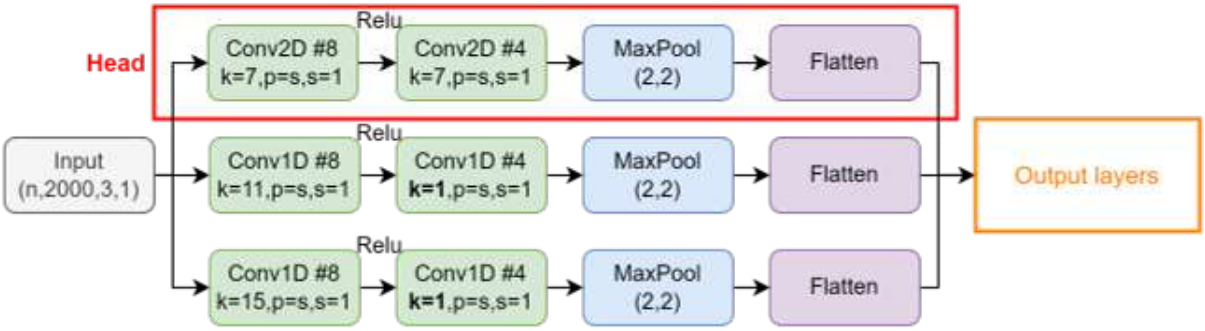


Figure 16: 2D Three-head Convolution

use the 'same' padding to avoid negative dimensions during training. Additionally, each convolutional layer generates fewer feature maps than the previous one. In addition, to lower the dimension, one extra convolutional layer with kernel size equal to 1 is placed in the second and the third convolutional heads. All of the remaining hyperparameters are consistent with the original Three-head CNN.

5.4.4 Embedded Deep Learning System

The nature of our system incentivised us to shift towards an embedded solution, which avoids the data transmission between the data source and main processor. In such a way, the privacy would be maximised, while the reliance on an external processor would be reduced. We exclude classic methods as they require substantial computing resources for data processing. The recent technological advancements has made feasible the deployment of deep learning models on resource-constraint device, such as microcontroller. As mentioned earlier, our chosen processor has built-in kernel library named CMSIS-NN, which could optimise some standard neural network operations (e.g., Convolution and Max-pooling) [54]. Although TensorFlow Lite offers us with a powerful toolkit for running deep learning models on the microcontroller, there are some significant constraints. Firstly, the model complexity was limited by the memory capacity of Arduino. Secondly, numerous operators in TensorFlow, including the `concatenate` used previously, are not compatible with TensorFlow Lite [55].As a result, not all models could be deployed on board. In addition to the inference accuracy, we evaluate the embedded system in terms of overall inference time, which is defined as the time spent from sampling data to obtaining the inference result (Equation 3).

$$T_{overall} = t_{transmission} + t_{inference} \tag{3}$$

The same test set is used for on-board performance evaluation. Samples are transmitted sequentially to the Arduino via serial protocol, and a predicted label is returned to the PC for each received sample. Then, the accuracy, and the average $t_{transmission}$ and $t_{inference}$ are calculated on PC.

6 Experimental Results

We conducted five experiments:

- The comparison between the traditional machine learning model and our proposed stacking machine learning model.
- The comparison of naive CNN and three-head CNN.
- The comparison between 1D CNN and 2D CNN.
- Classify between two, three, and four words.
- The result of using four subjects' data to train the model and using the last subject's data to test the model performance.

6.1 Experimental Setup

We evaluate six architectures in our experiments, including Random Forest, Staking Machine Learning Classifier trained with the augmented dataset, Naive 1D and 2D CNN, and Three-Head 1D and 2D CNN. It is worth noting that Naive CNN is a single-head convolutional neural network. We use an Adam optimizer for 40 epochs with a linear decay learning rate scheduler and a 5-epoch linear warm-up for deep learning algorithms. A batch size of 32 is employed, as well as an initial learning rate of 0.001 and a weight decay of 0.04. To avoid overfitting without compromising model accuracy, early stopping with patience = 5 (i.e., the number of epochs without a drop in loss) is applied. All tests are carried out at least five times on the Tesla T4 GPU processor to reduce randomness.

Further, we test the impact of challenging test samples on our networks. This section of experiment is inspired by the fact that some words involve similar movements and activation of muscles, which may result in signals difficult to be distinguish. We train the best model obtained in previous experiment with 2 words, 3 words and 4 words respectively. For 2 words situation, we use 'Eat' and 'Drink' or 'Write' and 'Say' as they are empirically simpler to classify. We then exclude 'Write' and retain the rest 3 words because it is observed that 'Write' and 'Drink' share similar signal patterns. At last, all 4 classes of words are included.

Last but not least, we examine the generalization ability of our best model in real-time use. The reason why we design this experiment is that the way of controlling muscles may vary between different subjects. Therefore, it is important to test how well our model can do on different subjects which may implicitly indicate its real-time performance. We define two training styles in this experiment: *Excluding One* and *Including All*. For *Excluding one*, we exclude the data from one of our subjects (subject 5) during training and use his/her data for test only. For comparison, in *Including All*, we train with data from all the subjects (reserving part of data from subject 4 as test set) and also test with the data from subject 4 only.

6.2 Evaluation Criteria

For evaluating different architectures, we analyse multi-class classifiers using weighted precision, weighted recall, and weighted F1-score for our suggested approaches. Each

evaluation criterion is mathematically specified as follows:

$$Weighted\ Precision = \frac{\sum_{i=0}^{k} T_i \cdot \frac{TP_i}{TP_i+FP_i}}{\sum_{i=0}^{k} T_i}$$

$$Weighted\ Recall = \frac{\sum_{i=0}^{k} T_i \cdot \frac{TP_i}{TP_i+FN_i}}{\sum_{i=0}^{k} T_i}$$

$$Weighted\ F1\text{-score} = 2 \times \frac{Weighted\ Precision \times Weighted\ Recall}{Weighted\ Precision + Weighted\ Recall}$$

Where TP signifies the number of true positives, FP the number of false positives, FN the number of false negatives, and T the number of true labels for each class.

For the evaluation of models on challenging samples and the evaluation of their generalization ability on real-time scenario, we use the prediction accuracy of our models as the criteria.

6.3 Results

6.3.1 Model Performance on PC

Table 2 shows the performance of both machine learning and deep learning approaches on PC. The confusion matrices are in the appendix figure 20.

Model	Accuracy (%)	F1-score	Precision	Recall
Random Forest	76.2	76.49	76.71	76.28
Stacking ML + Data Aug.	82.74	83.07	83.40	82.74
Naive 1D-CNN	77.38	77.82	78.27	77.38
Three-Head 1D-CNN	82.74	82.72	82.71	82.74
Naive 2D-CNN	78.57	78.61	78.66	78.57
Three-Head 2D-CNN	86.05	84.68	84.84	84.52

Table 2: PC Performance Evaluation

Test Words	Test Accuracy (%)
Eat, Drink	90.27
Write, Say	91.75
Eat, Drink, Say	89.38
Eat, Drink, Write, Say	86.05

Table 3: Performance of best model on different subset of words

Training Strategy	Test Accuracy (%)
Including All	86.05
Excluding One	31.55

Table 4: Test accuracy of model training with two training strategies

6.3.2 Model Performance on Board

Due to the fore-mentioned limitations, only the Naive 2D-CNN has been successfully deployed on board. Since the model already has a relatively low memory footprint, no quantisation is applied to preserved the performance. Table 5 compares the performance of Naive 2D-CNN on board against the performance on PC in terms of accuracy over the test set, average $t_{inference}$, and $T_{overall}$. Using Equation 3, the $T_{overall}$ can be calculated. The average $t_{transmission}$ for one sample is measured to be 2121 ms.

System Name	Accuracy (%)	Avg. $t_{inference}$ (ms)	$T_{overall}$(ms)
Original System	78.57	4	2125
Embedded System	78.57	687	687

Table 5: Performance comparison between original and embedded system in terms of accuracy and time efficiency

6.4 Discussion of Results

6.4.1 Effect of Stacking Generalisation and Data Augmentation

It is clearly observed from Table 2 that the ensemble machine learning classifier outperforms the random forest. Random forest model combines decision trees with bagging technique. Bagging technique on decision trees may not be powerful enough for complex patterns since the merging strategy is randomly determined by bagging, and it mainly focuses on overcoming overfitting instead of improving performance. On the contrary, for the ensemble model proposed in our project, logistic regression is applied to learn the combination of results from different models. This resembles a voting system to summarize the features and results provided by models, which is shown to be able to improve classification accuracy. In addition, augmentation techniques allow our model to have a stronger generalisation ability and yield lower test errors. However, as same as Random Forest, the stacking method also struggles to distinguish the words “Write” and “Drink”, due to their similar voice positions.

6.4.2 Effect of Multi-Head Structure

Experimental results show that the proposed multi-head CNN approach is promising in terms of the performance accuracy, recall, precision, and F1 score when compared with many other methods. In particular, compared to 1D single head CNN, utilising multi-head CNN architecture improves the classification accuracy on the test set by around 5%. A more significant improvement occurs as Three-Head 2D CNN is applied. The reason for this is that such a multi-head structure produces a decent fusion of features extracted

at different resolutions. Meanwhile, because the structure of each convolutional head is streamlined, as mentioned in Section 5.4.3, adopting this method will not result in an overly bulky model.

6.4.3 Effect of 1D Three-Head and 2D Three-Head

Observed from Table 2, using the three-head technique, 2D CNN is around 3.3% higher in accuracy and around 1.9% higher in F1-score than 1D CNN. This is probably because that 2D convolution introduces the kernel of size 1 combined with kernels of size 7. Such a combination enables the network to capture relatively long-term signal patterns and short-term ones simultaneously. Therefore, it is concluded that the 2D three-head CNN is the most suitable model for this project.

6.4.4 Model Performance on Simple and Challenging Words

The performance of our best model (i.e., Three-Head 2D CNN) varies depending on whether our wordset contains words with similar voice positions. As depicted in the figure. 3, our model performs well when discriminating basic two words (e.g., "Eat" and "Say"), and achieves accuracy beyond 90%. The test accuracy still retains around 89% when "Drink" is added to the subset. Nonetheless, when 4 words are trained together, the accuracy drops to around 86%. This is because our model becomes weak in distinguishing words that require similar muscle movements, constrained by the number of sensors. Specially, we observe that the sEMG signal for words "Drink" and "Write" are quite similar. This is also reflected in Table 3 where accuracy drops a lot when "Write" is added to training set. The most intuitive solution is to collect EMG signals from more distinguishable muscles but this will put huge burden on the subject's face and also require higher sensor precision.

6.4.5 Model Generalization Ability to New Subjects

From Table 4, we see that the model has a poor generalisation ability to samples from a new subject. The accuracy drops to sharply to 31.55% from 86.05%, indicating that the model may have not effectively learned the inherent patterns of the samples. Such a result may also originated from the external factors mentioned in 4.1. The muscle position variation between subjects may result in high dissimilarity between samples of same class. In other words, we are forcing the model to learn to differentiate object classes with large intra-class variation and potentially inter-class similarity.

6.4.6 Evaluation of Embedded System

As shown in Table 5, there is no difference in terms of accuracy between both systems. Such result aligns with the expectation, as the precision of model was preserved. In terms of average inference time, the original system (4 ms) outperforms greatly its counterpart (687 ms) due to its more powerful processor. However, In terms of overall inference time, the embedded system surpasses the original system, as it avoided the time for data transmission. It worth mention that the data transmission time is dependent on the communication protocol used. We did not evaluate the difference between protocols, as

it is out of project scope. The conclusion may be different if protocols with higher speed was used (e.g., Bluetooth).

7 Workplan

Long Term Plan:

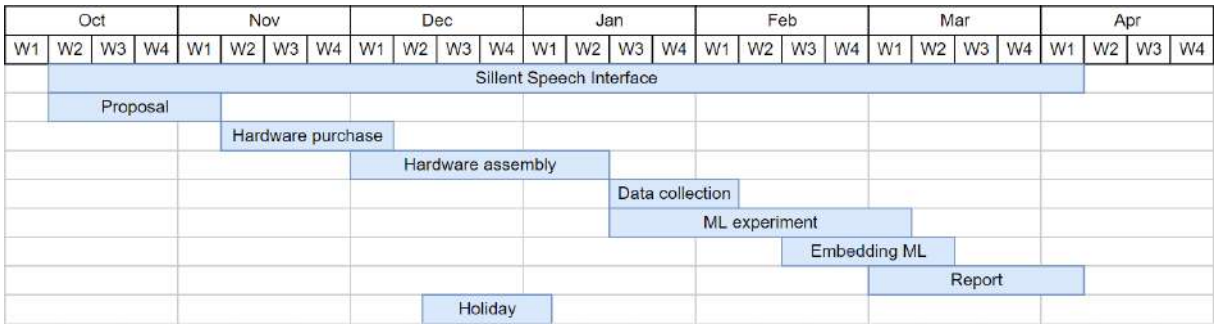


Figure 17: Gantt Chart of SSI project

Expense claim:

Components	Qty.	Unit price (£)	Total price (£)	Overall cost (£)
Laser cutter boards	4	5.19	20.76	930.69
Myoware sensors	15	34.9	523.5	
Arduino Nano33	3	28.5	85.5	
Cables	9	4.44	39.96	
Cable Shields	15	4.44	66.6	
Electrodes packs	20	7.14	142.8	
Sockets	5	5	25	
Stackable Headers	4	1.34	5.36	
USB cables	3	3.67	11.01	
Strip boards	6	1.7	10.2	

Table 6: Expensive claim of all hardware components

8 Conclusion

8.1 Summary

Our contributions in this project are typically four-fold.

- Use affordable sEMG sensors and Arduino to ensemble a silent speech interface to collect and construct a dataset.
- Apply cutting-edge signal processing techniques to the sEMG signals and extract usable features for the traditional machine learning model.

- Train several machine learning and deep learning models that can classify a few words and make fair performance evaluation on them.
- Deploy deep learning method on board, enabling real-time wireless inference.

8.2 Evaluation

As shown in the Table 6, we successfully build five functional prototypes within the budget. Moreover, we manage to create a stable and clean environment for data collection by utilizing sockets and acrylic cases, achieving hardware-level data denoising. In terms of the dataset, we collect in total 820 data for four subjects. After augmentation, 2460 reliable data is obtained, which is enough for training our model. We construct both traditional machine learning and deep learning models. Most importantly, we modify and enhance both models based on our understanding of the problem and achieve significant improvement in the performance. Thus, we conclude that all the objectives are met in a professional way.

However, we admit the drawbacks of our project. On the one hand, there are great limitations to the sensor. Though MyoWare is affordable, the resolution is not high enough for more complex tasks. A direct result of this limitation is that word choice significantly affects performance, and our model can only classify distinctive words. On the other hand, real-time performance is not satisfying. The preliminary reason behind this is the variability between different peoples. Though we try our best to find the general muscle locations that work for as many subjects as possible, data from different subjects still appears to have a great discrepancy since people all have their distinctive way of speaking. This leads to bad performance when the model is tested with unseen subjects.

Another major factors that lead to inconsistent model performance on the dataset and real-time test can be the repositioning of electrodes between trials. The sEMG signal may vary significantly if the recording electrodes are removed and reattached even for the same subject[18]. In fact, during our experiment, discrepancies between signals were also observed after the break, when the electrode were taken off and reattached. This deficiency was hardly improved on either the classical machine learning or the deep learning techniques. Time needed to be spent on calibrating the electrode positions until similar patterns were observed. With these two limitations, we proposed some future works.

8.3 Future Work

In the future, with some more precise sensors, it is possible to enlarge the word pool and decrease the interference of similar words. Recently, few-shot learning has been proposed to tackle the variance of EMG signals, handling the difference between subjects [56]. As aforementioned in Literature Review Section, [18], [19], and [20] also proposed possible solutions to this problem by adversarial training, transfer learning, and data adaptation, which unfortunately were not implemented in this project due to time limit. Thus, it could be a promising direction to extend our works to the session-independent class, where the disparity of accuracy between sessions is explored to be minimized.

Besides, the LSTM model is suitable for the sEMG-to-speech mapping. With LSTM, it may be possible to achieve the sentences classification and make the whole system be more suitable for actual circumstances for voice disorders.

References

[1] L. Breiman, "Random forests," pp. 5–32, 2001.

[2] J. R. Quinlan, "Induction of decision trees," pp. 81–106, 1986.

[3] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers: 2nd edition (with python examples)," 4 2020. [Online]. Available: http://arxiv.org/abs/2004.04523http://dx.doi.org/10.1145/3459665

[4] J. A. Gonzalez-Lopez, A. Gomez-Alanis, J. M. M. Doñas, J. L. Pérez-Córdoba, and A. M. Gomez, "Silent speech interfaces for speech restoration: A review," *IEEE Access*, vol. 8, pp. 177 995–178 021, 2020.

[5] Q. Wen, L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[6] D. R. Droual. Muscles of facial expression. [Online]. Available: http://droualb.faculty.mjc.edu/Lecture%20Notes/Unit%203/muscles%20with%20figures.htm

[7] M. Hussain, M. B. I. Reaz, F. Mohd-Yasin, and M. I. Ibrahimy, "Electromyography signal analysis using wavelet transform and higher order statistics to determine muscle contraction," *Expert Systems*, vol. 26, no. 1, pp. 35–48, 2009.

[8] A. D. C. Chan, K. Englehart, B. Hudgins, and D. F. Lovely, "Myo-electric signals to augment speech recognition," pp. 500–504, 2001.

[9] M. Fitzpatrick. Lip-reading cellphone silences loudmouths. [Online]. Available: https://www.newscientist.com/article/dn2122-lip-reading-cellphone-silences-loudmouths/

[10] G. S. Meltzner, J. T. Heaton, Y. Deng, G. D. Luca, S. H. Roy, and J. C. Kline, "Development of sEMG sensors and algorithms for silent speech recognition," *Journal of Neural Engineering*, vol. 15, no. 4, p. 046031, jun 2018. [Online]. Available: https://doi.org/10.1088/1741-2552/aac965

[11] M. Wand, C. Schulte, M. Janke, and T. Schultz, "Array-based electromyographic silent speech interface." in *Biosignals*, 2013, pp. 89–96.

[12] G. Buzsáki, C. A. Anastassiou, and C. Koch, "The origin of extracellular fields and currents—eeg, ecog, lfp and spikes," *Nature reviews neuroscience*, vol. 13, no. 6, pp. 407–420, 2012.

[13] B. Pesaran, M. Vinck, G. T. Einevoll, A. Sirota, P. Fries, M. Siegel, W. Truccolo, C. E. Schroeder, and R. Srinivasan, "Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation," *Nature neuroscience*, vol. 21, no. 7, pp. 903–919, 2018.

[14] B. Rodríguez-Tapia, I. Soto, D. M. Martínez, and N. C. Arballo, "Myoelectric inter-

faces and related applications: Current state of emg signal processing–a systematic review," *IEEE Access*, vol. 8, pp. 7792–7805, 2020.

[15] C. Jorgensen, D. D. Lee, and S. Agabont, "Sub auditory speech recognition based on emg signals," in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 4. IEEE, 2003, pp. 3128–3133.

[16] R. Netsell and B. Daniel, "Neural and mechanical response time for speech production," *Journal of Speech and Hearing Research*, vol. 17, no. 4, pp. 608–618, 1974.

[17] S.-C. Jou, L. Maier-Hein, T. Schultz, and A. Waibel, "Articulatory feature classification using surface electromyography," in *2006 IEEE international conference on acoustics speech and signal processing proceedings*, vol. 1. IEEE, 2006, pp. I–I.

[18] M. Wand, T. Schultz, and J. Schmidhuber, "Domain-adversarial training for session independent emg-based speech recognition." in *Interspeech*, 2018, pp. 3167–3171.

[19] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "A deep transfer learning approach to reducing the effect of electrode shift in emg pattern recognition-based control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 370–379, 2019.

[20] L. Maier-Hein, F. Metze, T. Schultz, and A. Waibel, "Session independent non-audible speech recognition using surface electromyography," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.* IEEE, 2005, pp. 331–336.

[21] X. Wang, M. Zhu, H. Cui, Z. Yang, X. Wang, H. Zhang, C. Wang, H. Deng, S. Chen, and G. Li, "The effects of channel number on classification performance for semg-based speech recognition," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).* IEEE, 2020, pp. 3102–3105.

[22] J. Freitas, A. Teixeira, and M. S. Dias, "Towards a silent speech interface for portuguese," *Proc. Biosignals*, pp. 91–100, 2012.

[23] S.-C. Jou, T. Schultz, M. Walliczek, F. Kraft, and A. Waibel, "Towards continuous speech recognition using surface electromyography," in *Ninth International Conference on Spoken Language Processing*, 2006.

[24] R. H. Chowdhury, M. B. Reaz, M. A. B. M. Ali, A. A. Bakar, K. Chellappan, and T. G. Chang, "Surface electromyography signal processing and classification techniques," *Sensors*, vol. 13, no. 9, pp. 12 431–12 466, 2013.

[25] Y. Deng, W. Wolf, R. Schnell, and U. Appel, "New aspects to event-synchronous cancellation of ecg interference: an application of the method in diaphragmatic emg signals," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1177–1184, 2000.

[26] C. Sinderby, L. Lindstrom, and A. Grassino, "Automatic assessment of electromyogram quality," *Journal of Applied Physiology*, vol. 79, no. 5, pp. 1803–1815, 1995.

[27] C. S. Pattichis and M. S. Pattichis, "Time-scale analysis of motor unit action potentials," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 11, pp. 1320–1329, 1999.

[28] P. Carre, H. Leman, C. Fernandez, and C. Marque, "Denoising of the uterine ehg by an undecimated wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 9, pp. 1104–1113, 1998.

[29] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "A comparative study of wavelet denoising for multifunction myoelectric control," in *2009 international conference on computer and automation engineering*. IEEE, 2009, pp. 21–25.

[30] K. Englehart, B. Hudgin, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 3, pp. 302–311, 2001.

[31] M. Hussain, M. B. I. Reaz, F. Mohd-Yasin, and M. I. Ibrahimy, "Electromyography signal analysis using wavelet transform and higher order statistics to determine muscle contraction," *Expert Systems*, vol. 26, no. 1, pp. 35–48, 2009.

[32] J. Rafiee, M. Rafiee, N. Prause, and M. Schoen, "Wavelet basis functions in biomedical signal processing," *Expert systems with Applications*, vol. 38, no. 5, pp. 6190–6201, 2011.

[33] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews™ in Biomedical Engineering*, vol. 30, no. 4-6, 2002.

[34] T. Schultz and M. Wand, "Modeling coarticulation in emg-based continuous speech recognition," *Speech Communication*, vol. 52, no. 4, pp. 341–353, 2010, silent Speech Interfaces. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167639309001770

[35] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, "A novel feature extraction for robust EMG pattern recognition," *CoRR*, vol. abs/0912.3973, 2009. [Online]. Available: http://arxiv.org/abs/0912.3973

[36] M. A. Oskoei and H. Hu, "Ga-based feature subset selection for myoelectric classification," in *2006 IEEE international conference on robotics and biomimetics*. IEEE, 2006, pp. 1465–1470.

[37] C. Altın and O. Er, "Comparison of different time and frequency domain feature extraction methods on elbow gesture's emg," *European Journal of Interdisciplinary Studies*, vol. 2, pp. 35–44, 2016.

[38] L. Maier-Hein, F. Metze, T. Schultz, and A. Waibel, "Session independent non-audible speech recognition using surface electromyography," vol. 2005, 2005, pp. 307–312.

[39] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[40] ——, "Statistical mapping between articulatory movements and acoustic spectrum using a gaussian mixture model," *Speech communication*, vol. 50, no. 3, pp. 215–227, 2008.

[41] V. Chandrashekhar, "The classification of emg signals using machine learning for the construction of a silent speech interface," *Credits & Acknowledgements*, vol. 4, pp. 2020–21.

[42] A. Abdullah and K. Chemmangat, "A computationally efficient semg based silent speech interface using channel reduction and decision tree based classification," vol. 171. Elsevier B.V., 2020, pp. 120–129.

[43] L. Diener, M. Janke, and T. Schultz, "Direct conversion from facial myoelectric signals to speech using deep neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–7.

[44] M. Janke and L. Diener, "Emg-to-speech: Direct generation of speech from facial electromyographic signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2375–2385, 2017.

[45] A. Albulbul, "Evaluating major electrode types for idle biological signal measurements for modern medical technology," *Bioengineering*, vol. 3, p. 20, 08 2016.

[46] K. S. Lee, "Prediction of acoustic feature parameters using myoelectric signals," *IEEE Transactions on Biomedical Engineering*, vol. 57, pp. 1587–1595, 2010.

[47] S.-C. Jou, L. Maier-Hein, T. Schultz, and A. Waibel, "Articulatory feature classification using surface electromyography," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1, 2006, pp. I–I.

[48] A. Kapur, S. Kapur, and P. Maes, "Alterego: A personalized wearable silent speech interface." Association for Computing Machinery, 3 2018, pp. 43–53.

[49] M. Pla Mobarak, G. J.M., M. R., and V. Louis-Dorr, "Transient state analysis of the multichannel emg signal using hjorth's parameters for identication of hand movements," 06 2014.

[50] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[51] Arundo. Time warp. [Online]. Available: https://tsaug.readthedocs.io/en/stable/_modules/tsaug/_augmenter/time_warp.html?highlight=time%20warp

[52] K. M. Rashid and J. Louis, "Window-warping: a time series data augmentation of imu data for construction equipment activity identification," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, vol. 36. IAARC Publications, 2019, pp. 651–657.

[53] Arundo. Add noise. [Online]. Available: https://tsaug.readthedocs.io/en/stable/references.html?highlight=noise#tsaug.AddNoise

[54] L. Lai, N. Suda, and V. Chandra, "Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus," *arXiv preprint arXiv:1801.06601*, 2018.

[55] TensorFlow. Tensorflow lite and tensorflow operator compatibility. [Online]. Available: https://www.tensorflow.org/lite/guide/ops_compatibility

[56] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S. F. Atashzar, and A. Mohammadi, "Fs-hgr: Few-shot learning for hand gesture recognition via electromyography," 11 2020. [Online]. Available: http://arxiv.org/abs/2011.06104

9 Appendix

9.1 Github Repository

Our Github link is https://github.com/laiwenq/AML_Lymsy.

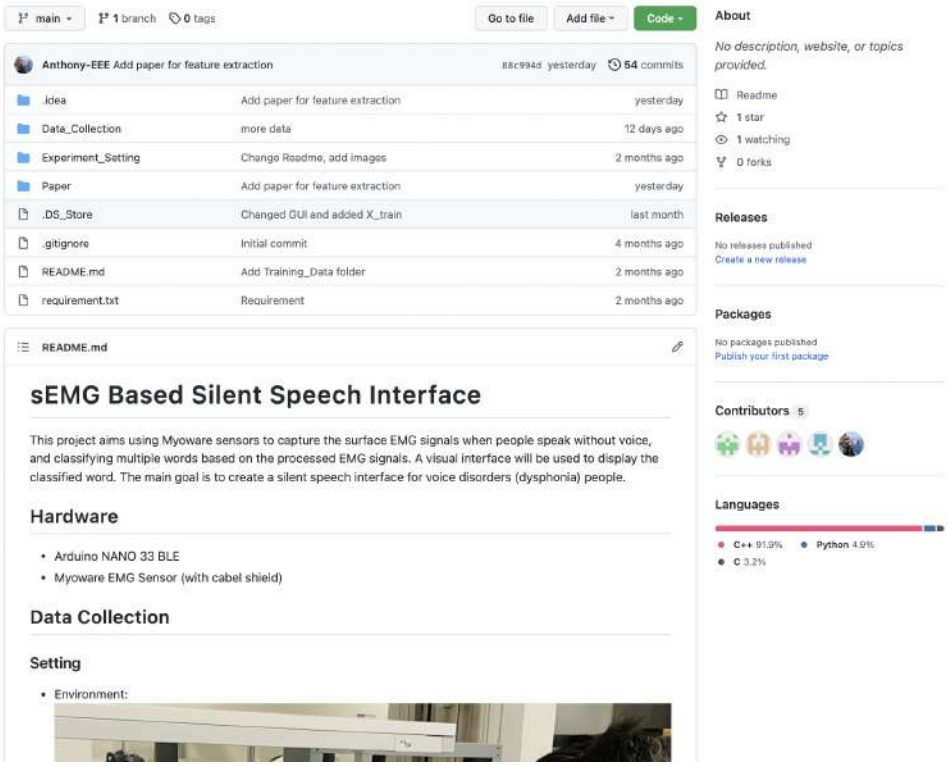


Figure 18: Github Readme Page

9.2 Hardware Screenshot

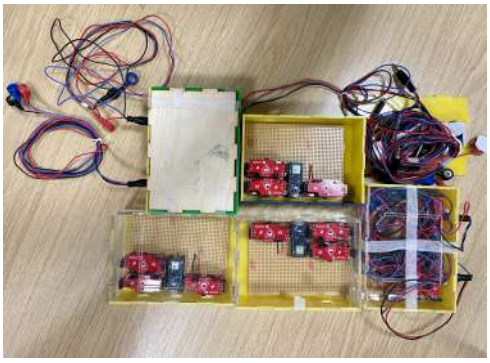
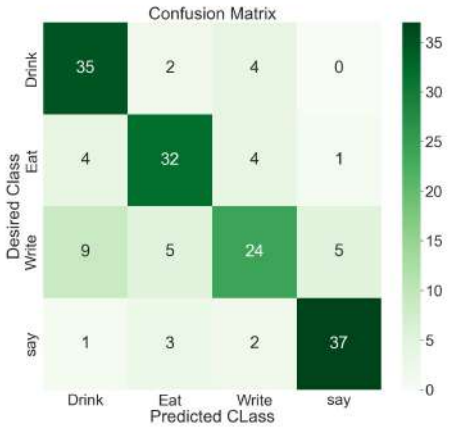
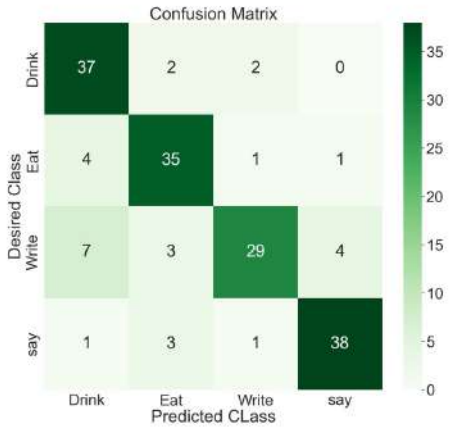


Figure 19: Hardware for 5 group members

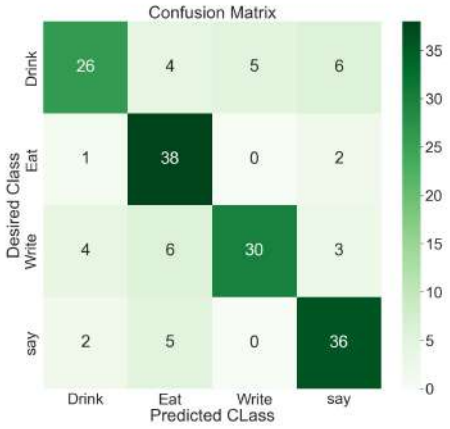
9.3 Confusion matrix for PC evaluation (Section 6.3.1)



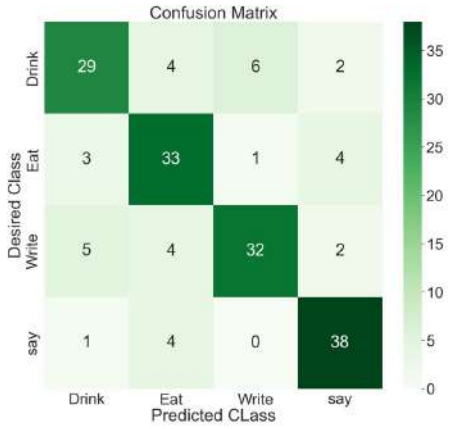
(a) Random Forest



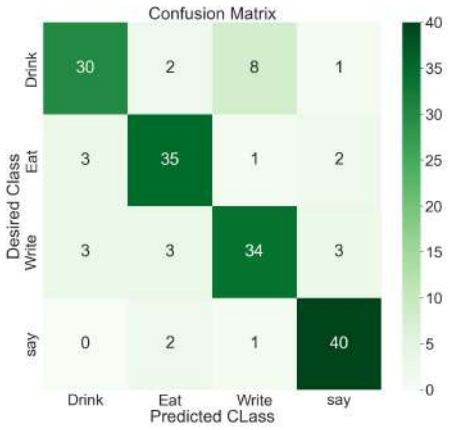
(b) Stacking ML Classifier



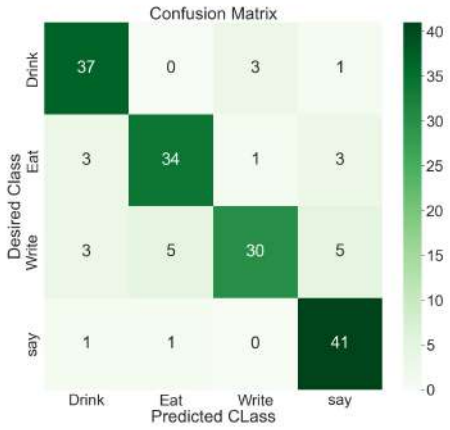
(c) Naive 1D-CNN



(d) Naive 2D-CNN



(e) Three-head 1D-CNN



(f) Three-head 2D-CNN

Figure 20: Confusion matrices of experimented models on PC