CSB2

FYP Report

Gesture to speech gloves

Lee Wing Piu, Sze Tak Ho, Kwong Chun Yiu

CSB2

Advised by

Prof. BENSAOU

Submitted in partial fulfillment of the requirements for CPEG 4901

in the

Department of Computer Science

The Hong Kong University of Science and Technology

2022-2023

Date of submission: April 20, 2023

Table of Contents

Introduction

1.1 Overview

Communication plays an important role in society, however, communicating with the hearing-impaired community could be hard because people seldom have exposure to sign language. With the help of machine learning techniques, much work had been done on classifying sign language with cameras. Nevertheless, it is not always good to use a camera in public as it could cause privacy issues. Therefore, this project will solely rely on motion sensors.

Thanks to the rapid development of sensors, microprocessors, and mobile apps, it becomes easier and easier to convert sign language to speech by utilizing IMU. However, not only do most IMU systems only consider a single gesture at a time, but they are also not mutable once deployed. This will be a great obstacle in communicating because people need to do a sentence of gestures instead of one gesture at one time in daily life. Therefore, this project will work on classifying a sequence of gestures from IMU devices, with a proposed extensible system that could be managed with an application.

Inertial measurement unit (IMU)

Due to the rapid development of IMU, 3-axis gyroscopes and accelerometers are available at a cheap price. Since they are compact and have low power consumption, they are very suitable for making portable devices. In this project, 5 MPU6050 modules which contain a 3-axis gyroscope, and an accelerometer will be used together with a Raspberry Pi to capture the hand gestures of users, then the data will be sent via Bluetooth for further processing.

Raspberry Pi Zero W

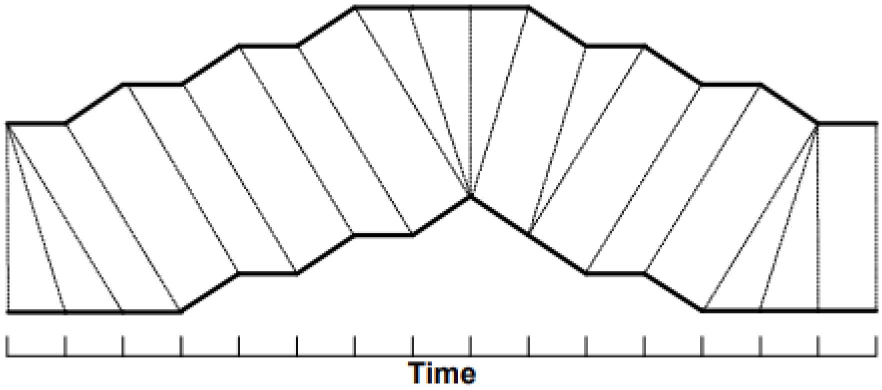
A single-board computer can perform most functions of a modern computer with all components soldered on a tiny size PCB. Thanks to its portability, it is suitable for many small-scale projects. A Raspberry Pi Zero W is used to record input data, preprocess the data, then send it to a smartphone via Bluetooth.

Gesture Recognition Framework

In general, there are two common ways of gesture recognition methods, using camera or motion sensors. In our project, we will solely utilize motion sensors due to privacy concerns. By utilizing the 3-axis accelerometer and gyroscope, we can detect the hand motions in each finger. This enables us to detect pre-defined meaningful gestures and give a suitable output by using a supervised machine learning model.

Velocity-independent design with dynamic time warping (DTW)

To make sure the system can give consistent output with different speeds of motion, the dynamic time warping algorithm will be applied. DTW will return a distance matrix that could align each point in a sequence to its nearest point in another sequence. Therefore, it will return an optimal “alignment cost” that could be used to tell the similarities between different sequences of data. It can classify gestures even with different speeds [5].

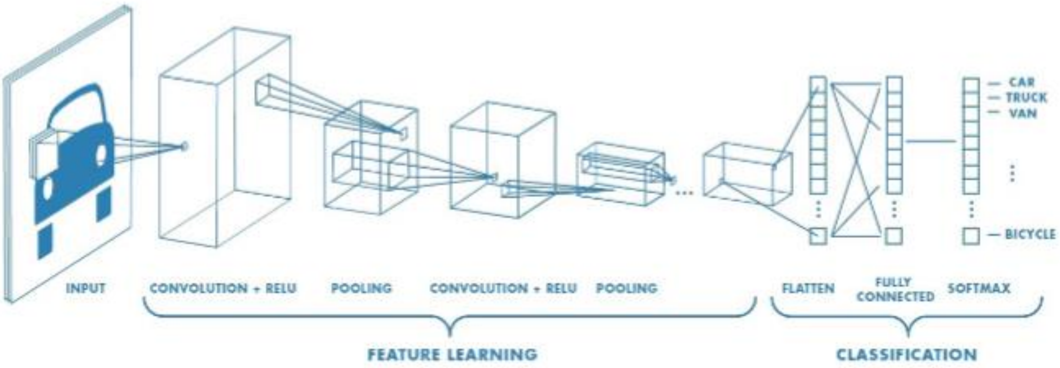


Gesture management application

Besides the glove, we will also develop a cross-platform application using Flutter for the text-to-speech (TTS) function and management of the glove. Once the application connects to the glove, it will receive the recorded gesture, then send it to a server for classifying the words or sentences. Finally, the application will display the results and play them in voice. The user can configure settings such as volume and pitch for the TTS.

Convolutional network

Convolution is a linear operation that applies a kernel into an image. This operation is sometimes called feature learning. By applying a kernel into data, the trainable weight is shared among data, therefore it will have significantly smaller amount of learnable weight compared to a fully connected layer. Thus, reduces training time and classifying time. After learning the features, tensors will be passed to the denser layer for classifying.



Source: COMP4211 lecture 06, Dit-Yan Yeung

1.2 Objectives

By converting hand languages into speech with gyroscopes, accelerometers, and a microprocessor, we hope to enhance the connection between the hearing-impaired community and general society. Some major milestones and challenges are listed to show the procedures to achieve our goal.

1. **Build a responsive data preprocessing channel** that could first continuously alleviate input errors from 5 gyroscopes and accelerometers. Then send the obtained processed data from Raspberry Pi to mobile phone.
2. **Train a supervised ML model** to classify the text to speak accurately, which can handle different speeds of performing hand gestures and different hand shapes.
3. **Develop a mobile application to pair with the glove as a control center** with functions including speaking the words or sentences classified, modifying the settings of the test-to-speech function, and storing the history of past recognitions.

1.3 Literature Survey

A lot of research has been conducted to capture and recognize gestures. This section reviews and analyses the existing solutions on gesture capturing and the algorithms of gesture recognition. There are two main approaches for translating sign language, which are vision-based and sensors-based systems. Although we are focusing on sensor-based systems, research on the other side can still give us some insights.

1.3.1 Vision-based recognition system

In 2014, Paulo Trigueiros, Fernando Ribeiro and Luís Paulo Reis developed a real-time Portuguese sign language recognition system with one Kinect camera. [1] The project utilized and compared two hand features for recognition, which are “Centroid distance” and “hand depth distance” respectively. The centroid distance method calculates the distance between the centroid and the boundaries to find the translation of the hand. The hand depth distance method records the distance between the object and the camera. In the end, these two methods achieved accuracies of 99.6% and 99.4% respectively with an optimized Support Vector Machine (SVM) model. While the accuracy is satisfactory, there are some heavy restrictions to the vision-based system, including the distance limitation of the camera, requirement of bare hand and the inability to function under sunlight.

1.3.2 Sensors-based Gesture Capture

To capture the complete gesture, finger bending, movement and orientation of the hand need to be measured.

In [2], 7 Bend sensors, 4 Hall Sensors and a 3-axis accelerometer are mounted on the glove. 4 bend sensors are placed on the proximal interphalangeal joints (PIJ) and 3 are placed in between the metacarpophalangeal joints (MCP) as shown in Figure 1. When the fingers are bent, the Hall effect sensors on the tips get close to the magnet placed on the palm (Figure 2). The output of the sensors becomes high. The orientation and movement of the hand are measured by the accelerometer. With bend sensors as the analog input and the hall sensors as the digital input, gestures can now be differentiated not only by the extent of bending but also the states of the Hall effect sensors. With logistic regression, the average accuracy of all gestures is 96%. Despite the higher precision achieved by the Hall effect sensors, those sensors placed on the fingertips hinder the usage of smartphone.

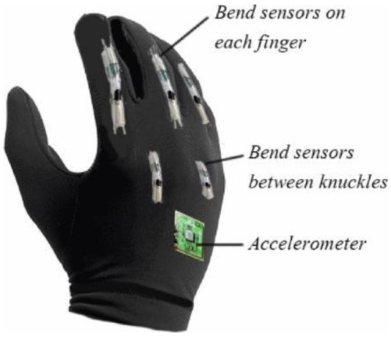


Figure1. Placement of bend sensors and accelerometer on the glove. Source: [2]

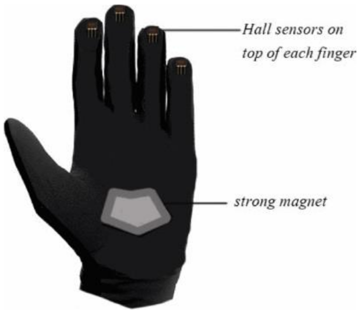


Figure2. Hall sensors on the fingertips and a magnet on the palm. Source: [2]

Another project used 1 accelerometer and 5 electromyography sensors to build the gloves. The EMG sensors will be capable of detecting the start and end point by detecting the intensity of EMG signal of muscle. A decision tree and hidden Markov models are used for classifying 72 Chinese Sign Language words [3].

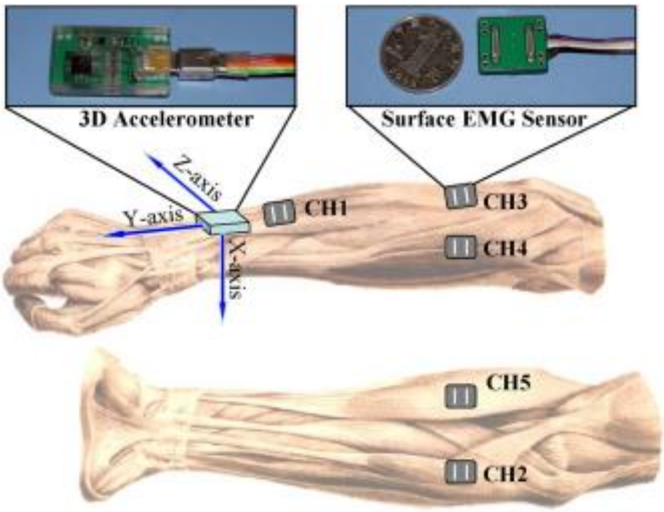
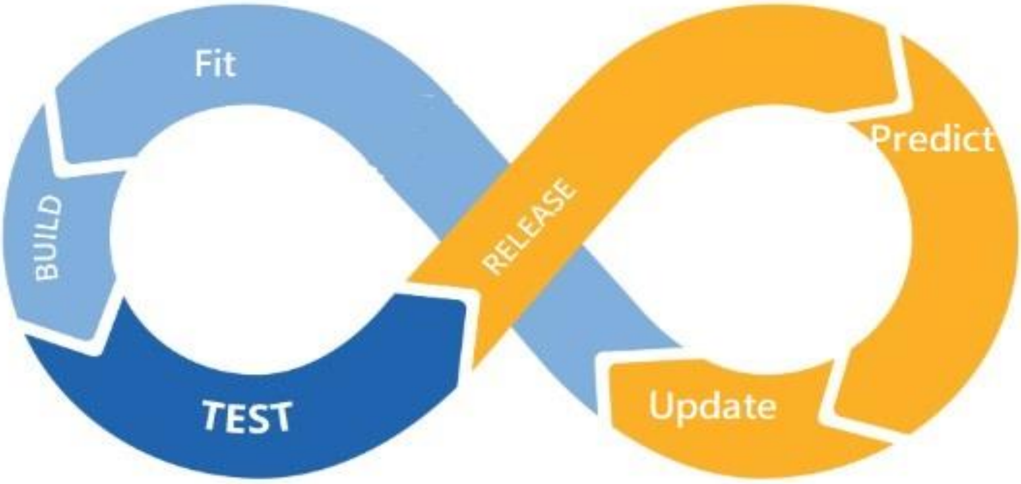


Figure3. a system with 4 EMG sensors with 1 accelerometer. Source: [3]

In this project, different from the previous projects, only one type of sensor module will be used to reduce complexity. A total of 5 MPU6050 which include both accelerometer and gyroscope will be used for capturing motions. Instead of classifying one gesture at a time, the system will be classifying a sentence at a time. Also, instead of using a fixed size of gesture data database, the system will allow users to define their own gestures by recording the gesture using an application.



2 Methodology

2.1 Design

This project is separated into 3 parts, namely the data fusing pipeline, the classifying model, and the application. The data fusing pipeline is responsible for denoising and sending the data to the application. Then, the classifying model will locate meaningful gestures and turn them into human language. The application hooks everything into a user interface, allowing the user to control the system.

The data-fusing pipeline

2.1.1 5 sensors multiplexing data input design

In the gloves, there are 5 sensor modules in total. To read the sensor values from all of them, we need to do multiplexing by ourselves. This is because there are not enough channels to allocate each sensor to a separate communication channel. To read from all 5 sensors consecutively, we need to switch on one sensor at a time and record their values one by one. As the frequency of the I2C channel is 200Hz, it is expected each of the sensors can have a sampling rate of around 40Hz which could give responsive output.

2.1.2 Kalman Filter noise reduction design

To reduce misalignments and shifts of the sensors, Kalman Filter will be applied as a noise reduction algorithm to make sure there will be no serious errors when operating time increases which can result in garbage-in garbage-out. The Kalman Filter will return a posterior estimate, which combines a prior estimate and a weighted difference between the actual measurement and a measurement prediction [4]. If the measurement error covariance is small, the actual measurement value is trusted more so it contributes more than the previous value; and if the measurement error covariance is high, the previous value is trusted more.

$$\underbrace{\hat{x}_k}_{post\ estimate} = \underbrace{\hat{x}_k^-}_{prior\ estimate} + \underbrace{K_k}_{Kalman\ gain} (\underbrace{Z_k}_{measurement} - H\hat{x}_k^-)$$

The classifying model

2.1.3 The first AI: 1-D Convolutional model for single gesture classification

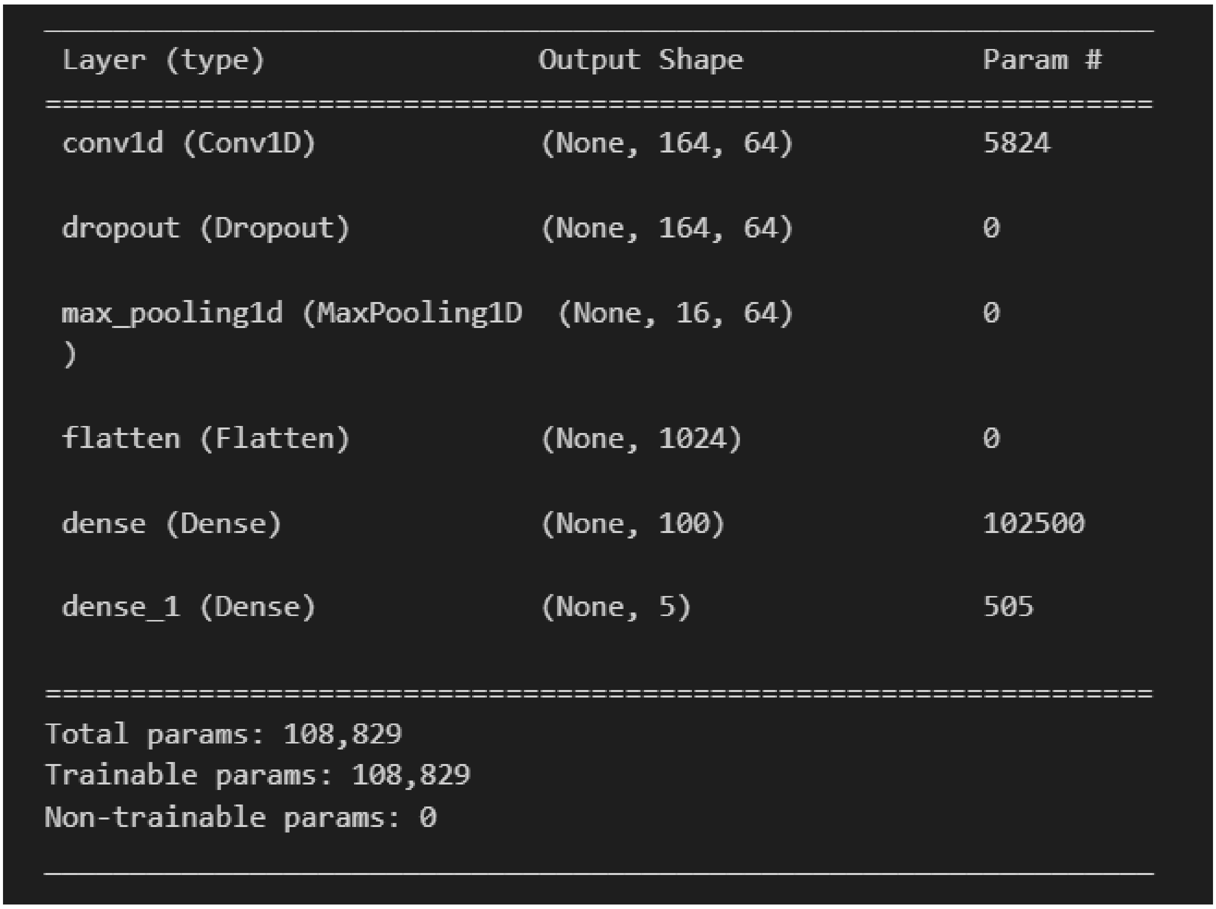
<u>Initial approach: KNN-DTW</u>

To kick-start our project, we first implemented a model for classifying one gesture. The implementation of KNN is popular, however, most of them use the Euclidean distance or Manhattan distance as the distance formula. This did not work out for us because those algorithms can only handle discrete data but not time-series data. In other words, they are for 1-dimensional data, but we need to classify 2-dimensional data. In our classifier, we used the Dynamic Time Warping algorithm as the distance formula to calculate a cost matrix between each gesture. It then returned the minimum cost to align the two gesture signals. Not only does this allow us to perform dimensionality reduction on time series data, but it also neglects the effect of doing a gesture fast or slow [8].

<u>Current approach: CNN</u>

One drawback of the KNN-DTW approach is that it runs slow. This affects the overall quality of the product by reducing responsiveness. Therefore, we investigated how to apply faster model on the problem. The final solution is to simply use a one-dimensional convolution model, which would allow us to avoid the expensive pair-wise comparisons.

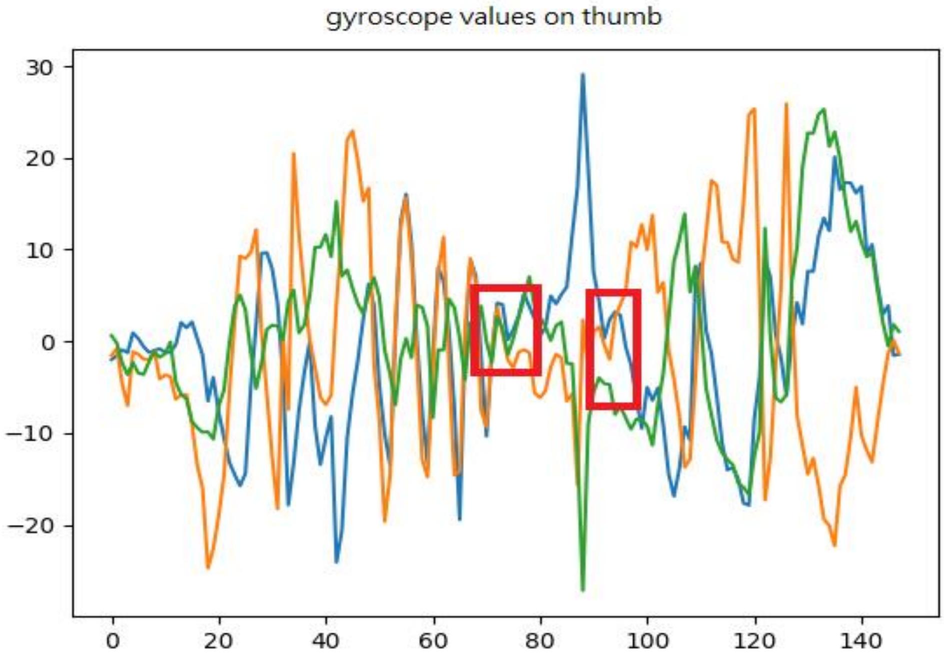
The model consists of one 1-D convolutional layer, one drop-out layer, one 1-D pooling layer, one flattens layer, and finally 2 fully connected layers. Thanks to the weight sharing property of a convolutional network, trainable weights are significantly reduced. Thus, deeper network can be applied which will increase representation power of the model. The model finally consists of 108,829 trainable weights, which is quite decent compared to other deep networks.

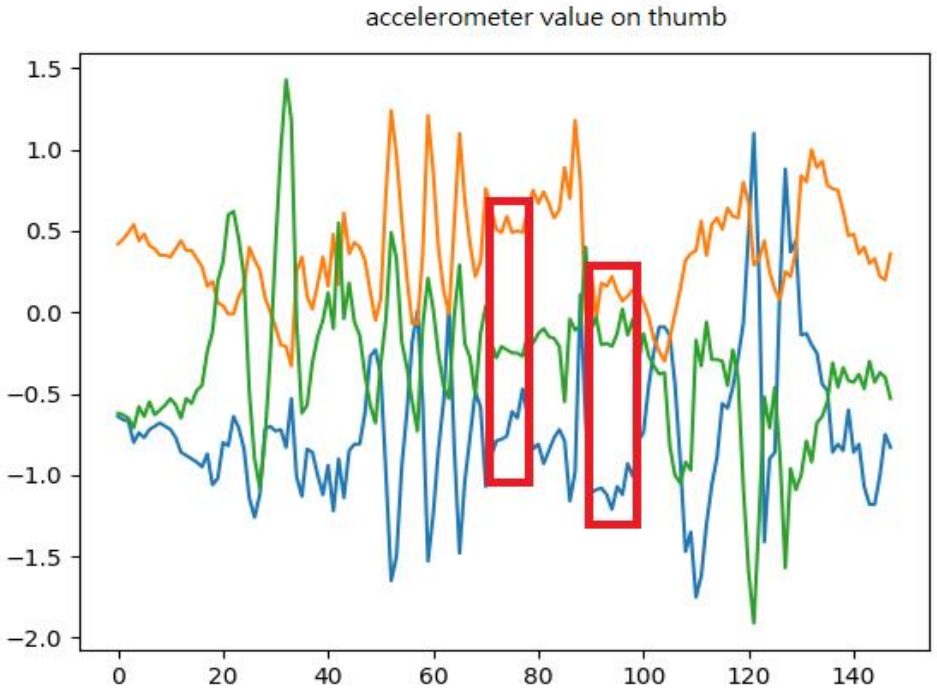


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 164, 64)           5824

 dropout (Dropout)           (None, 164, 64)           0

 max_pooling1d (MaxPooling1D  (None, 16, 64)           0
 )

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 100)               102500

 dense_1 (Dense)             (None, 5)                 505

=================================================================
Total params: 108,829
Trainable params: 108,829
Non-trainable params: 0
_________________________________________________________________
```

2.1.4 The second AI: Sliding window for breakpoint identification

To break a sequence of gestures into separate gestures for our model to handle one by one, we applied a sliding window algorithm. Inside each window, the variance of each column of data is calculated. Then we produced a score based on the variance of the gyroscope columns and the variance of the accelerometer columns. If each of the scores is significantly below its neighbors, we define the window as a breakpoint of gestures. This is because, between each gesture, we observed that there tends to be a temporary pause. If the window contains a pause, the variance will be significantly lower. This idea comes from the skill of NLP. Below is an example of the figure of the sensor values of one finger with a red box indicating a breakpoint.





2.1.5 Start point and end point detection

To tell the machine when to start processing the data, some fixed rules are needed. The easiest and the most conventional way is to perhaps press a button to indicate a starting point and an ending point. The benefit of this simple approach is that it saves computation resources for checking the start and end points. Only one GPIO input is checked instead of all sensors which could save battery usage in a portable device.

The mobile application

2.1.6 Application Design

The application serves a straightforward purpose, which is to act as a bridge between the gloves and the AI and display the result. The flow is as follows:

1. The application signals the glove to start recording
2. The application signals the glove to stop recording and transmit the data via Bluetooth

3. The application passes the data to the AI, displays the result, and plays TTS.

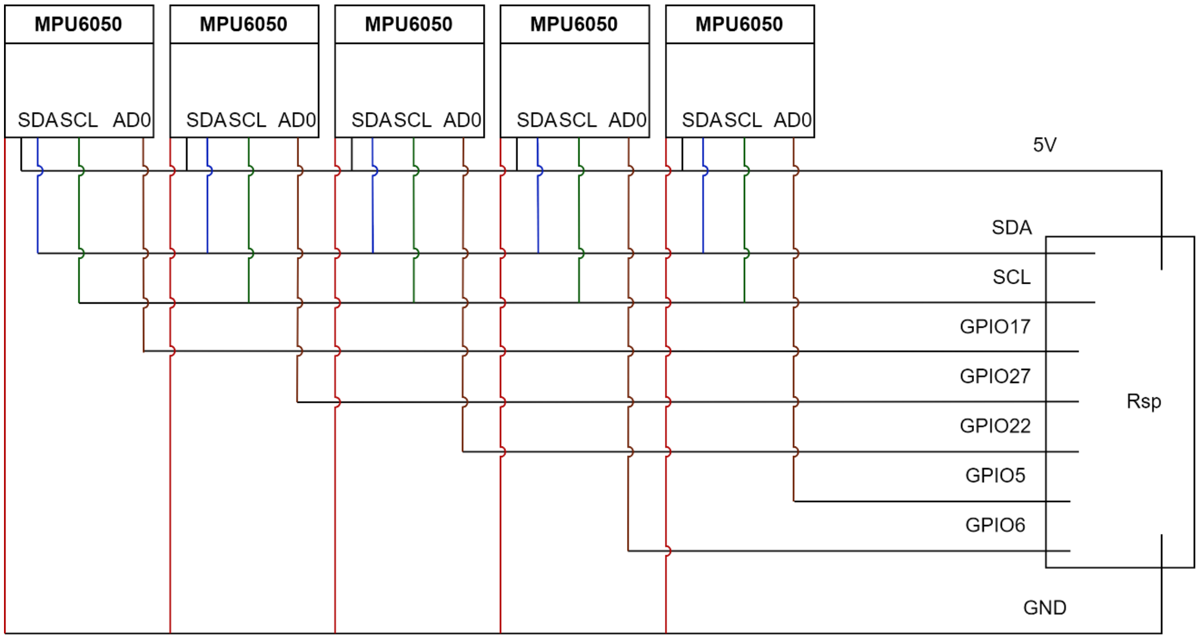
The Bluetooth connection is required to be secured first, which will then lead to the main page with the result display and other settings.

2.2 Implementation

The project is implemented in 3 stages. One person coordinates the hardware and provides the filtered sensor data continuously via Bluetooth. Another person uses pre-recorded data to implement different models for classifying a sequence of gestures into a meaningful sentence. The final person prepares an application that can take the data, put the data into a model, and say the sentence.

The hardware

2.2.1 5 sensors multiplexing data input implementation



As there is only one I2C bus in raspberry pi zero, we did multiplexing ourselves to read from 5 sensors consecutively. We connected all 5 sensors in parallel and connected the circuit to the I2C bus. In each loop, we output a HIGH voltage with a GPIO pin from the RSP to the AD0 pin in MPU6050. This switched the address of the AD0 pin from 0x69 to 0x68, in other words, ‘woke’ the sensor ‘up’. The data of that particular chip was output to the I2C channel to be further processed. In our implementation, the sampling rate of the data is approximately 40Hz which is high enough to display details of each gesture.

2.2.2 Kalman Filter noise reduction implementation

In general, the implementation of the Kalman filter could be divided into 2 parts, the time update part and the measurement update part. The time update part (or in other words the “predict” part) is to bring forward the current state and error covariance by 1 unit time to obtain the prior estimate for the next calculation. The measurement update part (or in other words the “correct” part) is to obtain an improved posterior estimate by incorporating a new measurement into the obtained prior estimate. The beauty of the computation is that the two parts can be computed recursively, which means the new state only needs to consider one previous state without needing to consider all the previous states. Note that the variable will be in an n*n matrix (list) so we can import “NumPy” for matrix multiplication and addition to make matrix computation less tedious.

Time Update (**"*Predict*"**)

(1) *Project the state ahead*

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

(2) *Project the error covariance ahead*

$$P_k^- = AP_{k-1}A^T + Q$$

***Measurement Update*(Correct)**

(1)*Compute the Kalman gain*

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

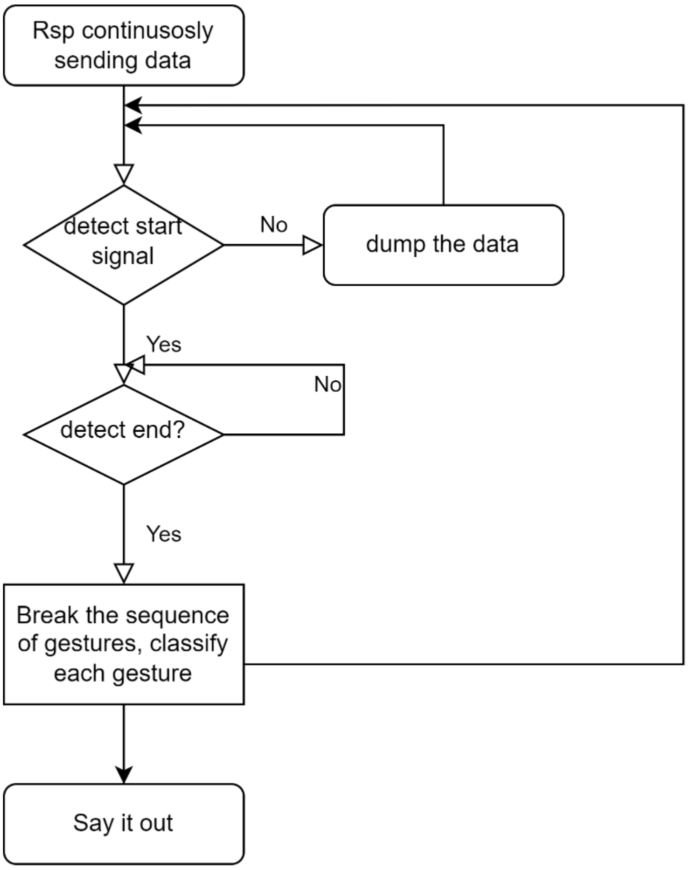
(2)*Update estimate with measurement* z_k

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$

(3) *Update the error covariance*

$$P_k = (I - K_kH)P_k^-$$

The pipeline


Rsp continusosly sending data
detect start signal
No
dump the data
Yes
No
detect end?
Yes
Break the sequence of gestures, classify each gesture
Say it out

2.2.3 Start point and end point detection

In the design of the project, we want to define a gesture to serve as a gesture to signal the start of a sentence. In our implementation, pointing a finger to the sky is our definition of start. By completing the model, this part was implicitly completed. Because its implementation only required using the classifier of one gesture.

Although the above idea sounds nice, the reality is that the system will not be able to locate the above gesture. In other words, the system has to scan through all the entire sequence of signals, classify all the entire sequence of signals in order to capture a start point. This will be way more than what we can afford in terms of delay and responsiveness.

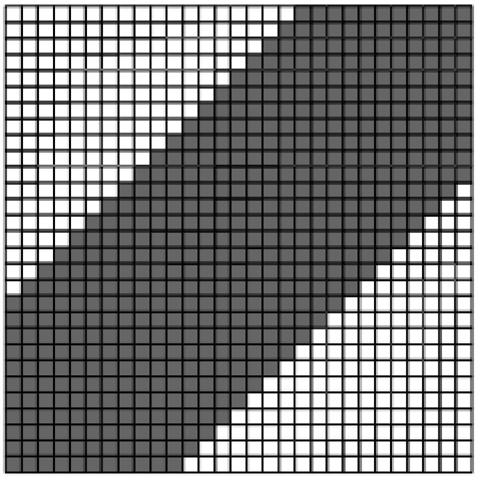
To reduce computational load, we decided to go back to pressing button, because it gives us the responsiveness that almost no other method can give us.

2.2.4 The first AI: 1-D convolutional model for single gesture classification

Our first approach is to use a KNN-DTW network. But it turned out to be too slow. Since this also gives us more than 90% accuracy, we would like to include its details here.

Past approach: KNN-DTW

The implementations of DTW varied a lot. Our best approach had reached a time complexity of O(n^2). However, in the paper of Stand Salvador and Philip Chan, they optimized the algorithm such as limiting the number of cells that are evaluated in the cost matrix, which lead to an almost O(n) implementation. Before adopting their implementation, it required 4 seconds to classify 3 gestures in a sequence. And it was improved to 3 seconds with their algorithm. (In the case of no multithreading)



(Grey area is the allowed warping window)

In the implementation of the KNN-DTW model, we first set up a Class so that we could use it as a module. The class imported Numpy and Scipy for matrix and statistics computation. The class can take 2 arguments in its instantiation, namely the number of neighbors to use for KNN and the maximum warping window allowed for DTW. In total, we defined 4 functions inside the class (excluding the constructor).

DIST_MATRIX ()

To classify a gesture, we first created a function to calculate the distance matrix between the testing dataset and the stored dataset. The pseudocode is as follows:

```
Create a M × N matrix, where M, N are the length of the testing and training dataset
for i ← 1 to M:
   for j ← 1 to N:
      matrix[i, j] ←fuse ( testing Dataset[i] , training Dataset[j] ) //see below
```

FUSE ()

The distance between a testing dataset and a training dataset is found by fusing the distance between all 30 columns. Since gyroscope values are measured in degree/s, the cost will be much higher compared to the values of the accelerometer which are measured in a fraction of G. Therefore, we need to normalize it to make the two values comparable. In extreme cases, if the normalizing factor is 1, the cost of the gyroscope dominates; if the factor is 0, the cost of the accelerometer dominates. By setting the factor in a range of 10-30, it gave different results based on the difference of similarities in the two types of meters.

```
for i ← 1 to 5
  testingsetMeters = testingset[i], trainingsetMeters = trainingset[i]
  for j ← 1 to 3
    cost_of_gyroscope.append(Fastdtw(testingsetMeter[j], trainingsetMeter[j]))
  for j ← 4 to 6
    cost_of_accelerometer.append(Fastdtw(testingsetMeter[j], trainingsetMeter[j]))
Return sum(cost_of_gyroscope) × normalizing factor + sum(cost_of_accelerometer)
```

FIT ()

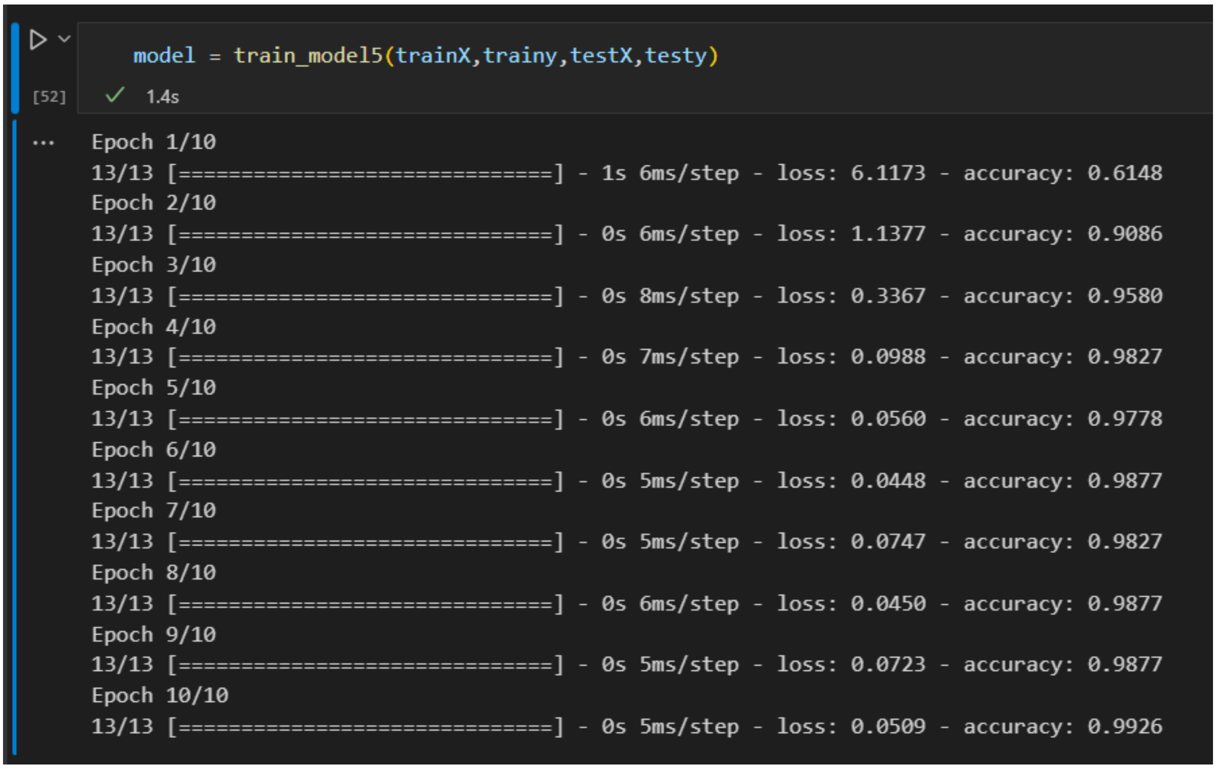
To fit the training data into the model, we defined the function ‘fit ()’, which is solely for assigning data and labels to the class.

PREDICT ()

To classify a gesture, we defined the function ‘predict ()’, which will take timeseries data and return a label. It first calls the dist_matrix() function, then sorts each row, and finally maps the K-nearest-neighbors' data to their label and returns the most referred label.

Current approach: 1-D Convolutional network

In our implementation of the convolutional network, we used TensorFlow with Keres Api.



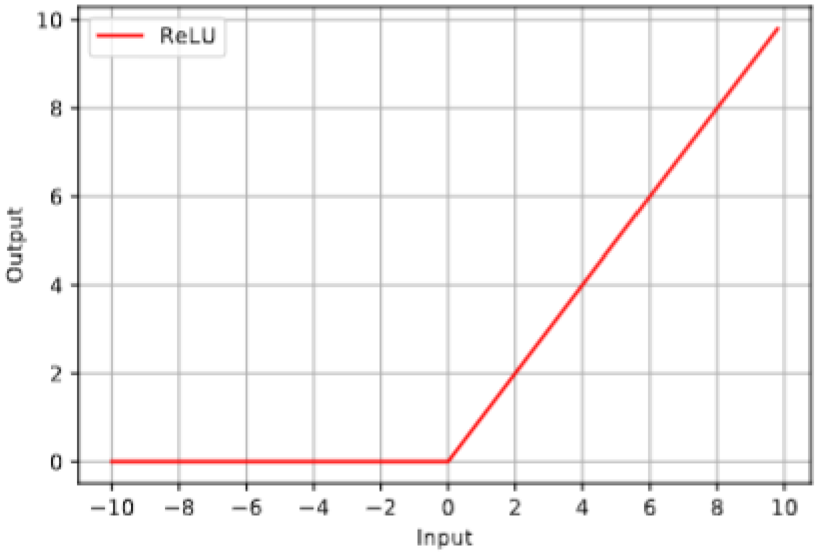
```
model = train_model5(trainX,trainy,testX,testy)
```

```
Epoch 1/10
13/13 [==============================] - 1s 6ms/step - loss: 6.1173 - accuracy: 0.6148
Epoch 2/10
13/13 [==============================] - 0s 6ms/step - loss: 1.1377 - accuracy: 0.9086
Epoch 3/10
13/13 [==============================] - 0s 8ms/step - loss: 0.3367 - accuracy: 0.9580
Epoch 4/10
13/13 [==============================] - 0s 7ms/step - loss: 0.0988 - accuracy: 0.9827
Epoch 5/10
13/13 [==============================] - 0s 6ms/step - loss: 0.0560 - accuracy: 0.9778
Epoch 6/10
13/13 [==============================] - 0s 5ms/step - loss: 0.0448 - accuracy: 0.9877
Epoch 7/10
13/13 [==============================] - 0s 5ms/step - loss: 0.0747 - accuracy: 0.9827
Epoch 8/10
13/13 [==============================] - 0s 6ms/step - loss: 0.0450 - accuracy: 0.9877
Epoch 9/10
13/13 [==============================] - 0s 5ms/step - loss: 0.0723 - accuracy: 0.9877
Epoch 10/10
13/13 [==============================] - 0s 5ms/step - loss: 0.0509 - accuracy: 0.9926
```

The model converges at superior speed. Achieving an accuracy of over 90% after the second epoch. We chose ‘Adam’ as our gradient descent method to do back-propagation.

Convolutional layer

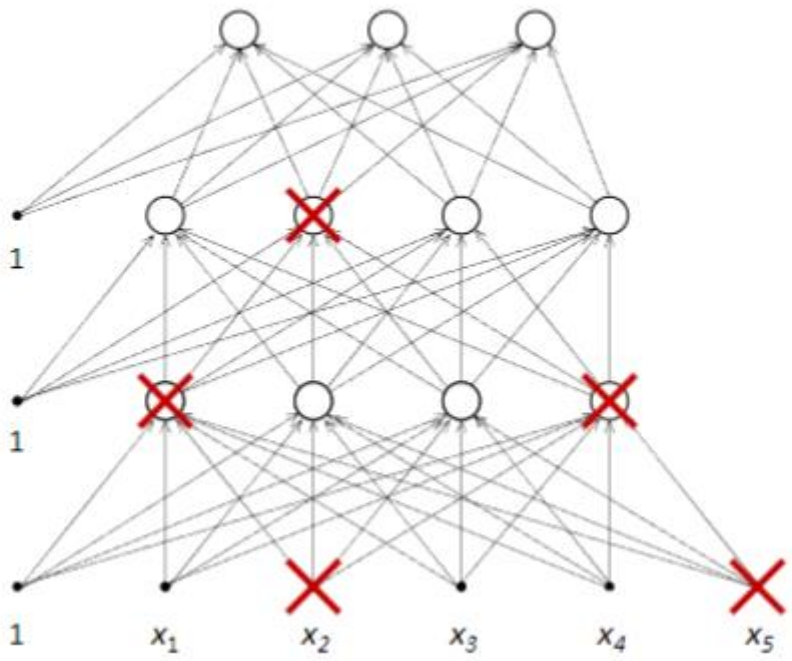
The convolutional layer consists of 64 output feature maps. Each with a kernel size 3. The activation function we chose is the rectified linear unit ‘relu’, defined as $f(x) = \max(0, x)$.



Source: 'RELU' function COMP4211 lecture 05, Dit-Yan Yeung

Dropout layer

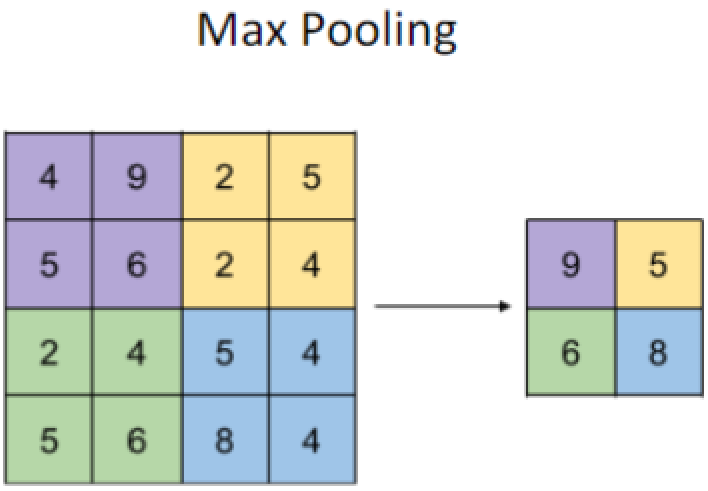
A dropout layer with a dropout rate of 0.5 is introduced to prevent overfitting in model training. The drop out layer is to make the network more robust. Since hidden units and inputs are dropped out randomly, it is similar to adding some noise to the input. This makes the network rely on more units or inputs and spread out the weights to more units.



Source: COMP4211 lecture 05, Dit-Yan Yeung

Max pooling layer

The pooling layer subsamples the input to a smaller input. This can reduce the computational load, memory usage and the number of trainable parameters. This layer consists of no weights but only an aggregation function. In our case we chose the max function.



4	9	2	5
5	6	2	4
2	4	5	4
5	6	8	4

9	5
6	8

Flatten layer

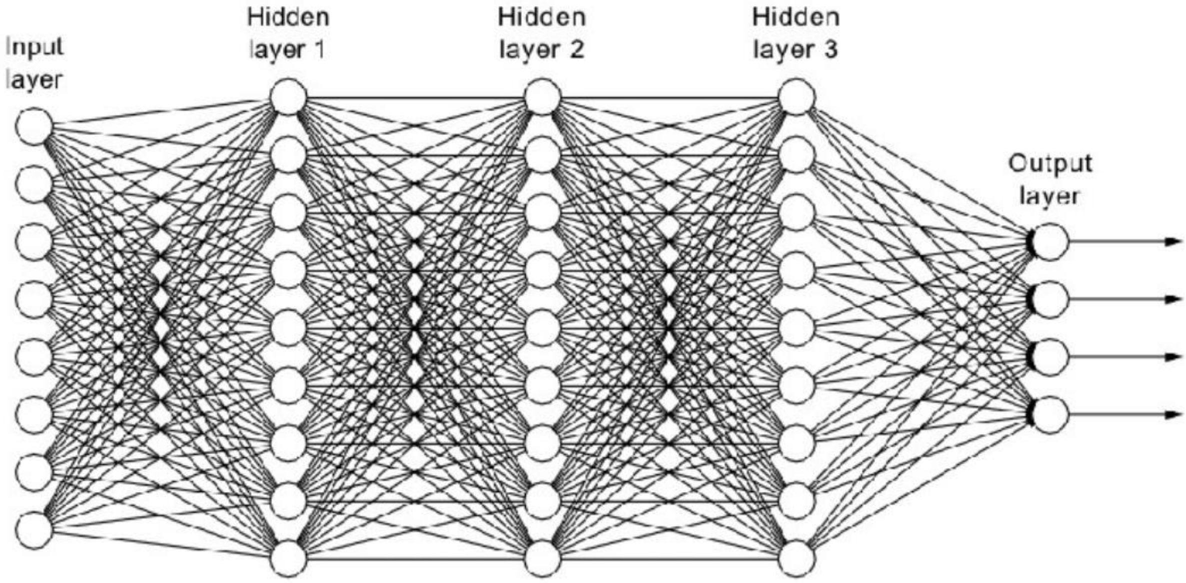
The flattened layer reshapes the tensor into a one-dimensional array, which is a step before classifying.

Dense layer

The dense layer is a fully connected layer for classifying purposes. We chose to put 100 computing units in this layer, with the 'relu' activation function.

Dense output layer

At the end, the tensor will pass through this layer to output its one-hot encoded label. Since this is a multiclass classification, a SoftMax function is applied to turn input into one-hot encoded output.



Source: researchgate.com Ying Da Wang

2.2.5 The second AI: Sliding window for breakpoint identification

To break a sequence of gestures into separate gestures, we wrote another Class for applying the sliding window algorithm and truncating gestures. The class is mainly composed of 2 functions. One is responsible for calculating the variances of the data inside the windows. The other one is to locate the index of the found breakpoint and return a list of separated gestures inside a sequence.

CHOP_DATA_INTO_FRAMES ()

The first function will slide a window into the timeseries data, then calculate the variance of each column vector. Finally, take the average of them and return it as a gyroscope variance list and an accelerometer variance list.

```
frames ← returned_list_from_sliding_window_algorithm
for i ← 1 to length(frames) // loop on each frame
    for j ← 1 to 5 // loop on each finger
        calculate the variance of each column, append to a list
        sum up the row of variance , append it to the variance list
return the variance list
```

SEPARATE_GESTURES ()

This function sees whether the variance of a window exceeds a threshold, if so, adds a breakpoint to the index. The indices will be used to break the sequence of data into parts. And the parts will be classified in parallel.

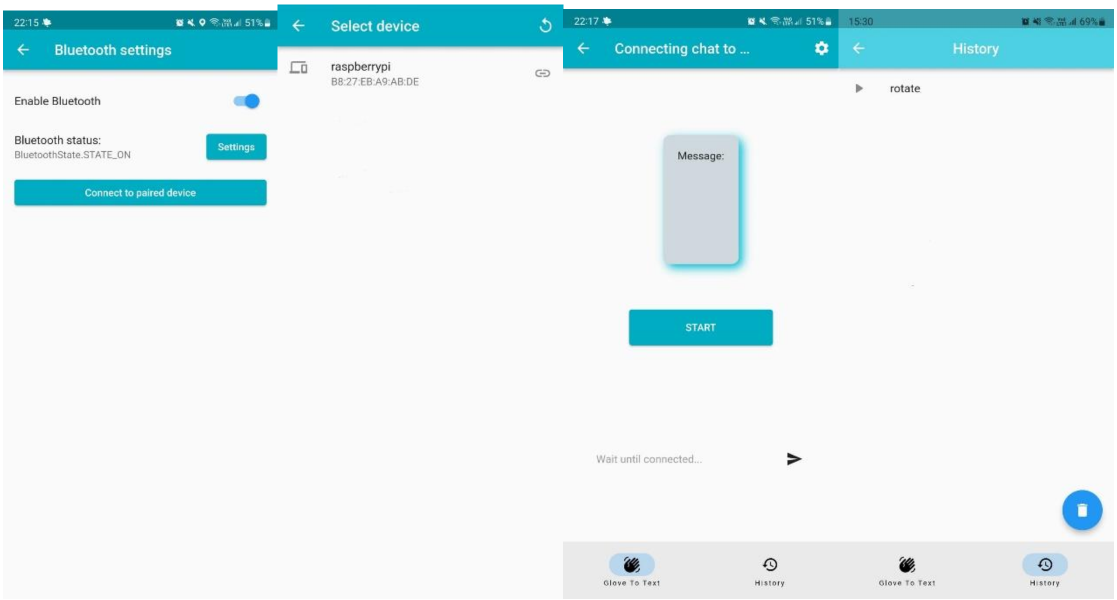
```
for i ← length of the list of variances
    if (variance score significantly lower than neighbors)
```

```
        append i to the break list
      else
        continue
    separate the sequence according to the calculated index
    return the separated list of gestures
```

The application

2.2.6 Application/Connection implementation

The application is developed using Flutter since it has a lot of convenient packages and the "hot-reload" feature. The Bluetooth connection is set up using the PyBluez module for Python in the Raspberry Pi and the Flutter Bluetooth Serial package for the application. The Flutter Blue Plus package provides features such as listing bonded devices and data transfer. Several useful example implementations of the package have been incorporated with the application [6]. The data transfer with the server is done using the built-in HTTP package from Dart.



From left to right, Page 1 - 4

There are four pages that the application mainly features.

Page 1 is the settings page that allows the user to view and configure Bluetooth settings before connecting the devices. Clicking on the “Connect to paired device button” leads to Page 2.

Page 2 is the paired device selection page. The names and MAC addresses of the paired devices are displayed to be selected, with a refresh button at the top right. Clicking on any of the devices leads to page 3.

Page 3 is the device interaction page. The title of the top bar displays the status of the current connection. The button in the middle is used to signal the glove to start recording and stop recording. When the Raspberry Pi sends the recorded sensor values to the application, the transition is split multiple times. The application then integrates the data and performs a POST request to the server. The returned result is finally displayed in the box above the button and a TTS is played. An input field is available near the bottom for manual typing TTS and a settings button is available at the top right for volume, rate, and pitch. The bottom is a navigation bar between this page and the history page.

Page 4 is the history page. This page stores all the past classified words and sentences with a clear button at the bottom right. Clicking on any item plays a TTS.

2.3 Testing

While we were testing the DTW-KNN model, we collected 5 samples from each of the 10 gestures, this is just a very brief testing because we were still proofing our concept. Unfortunately, this method ended up being too slow. When one full round of code was run in an Intel i7-9700k CPU, it took 3 seconds to finish. Therefore, we decided to find another method and gave up the KNN idea.

While we were testing the convolutional model, we collected 5 different gesture samples from different group members. Each gesture contains at least 120 samples. While the majority of the samples were connected by one member. Testing was conducted by combining the gesture samples from the remaining 2 members. Since all of us have different shapes and sizes of hands, it allows us to test the capabilities of our model in handling more difficult situations, and therefore can be proven to be more versatile. Since we can avoid pair-wise comparison, we can achieve much better computational speed. By applying batch processing, it only took less than 10ms to predict one gesture, when we were running it in the same setup.

2.3.1 Test the correctness of sensor data

To ensure the reliability and accuracy of the data obtained from the MPU-6050, it is crucial to thoroughly test and analyze its performance under various conditions. The output data of the MPU-6050s are evaluated.

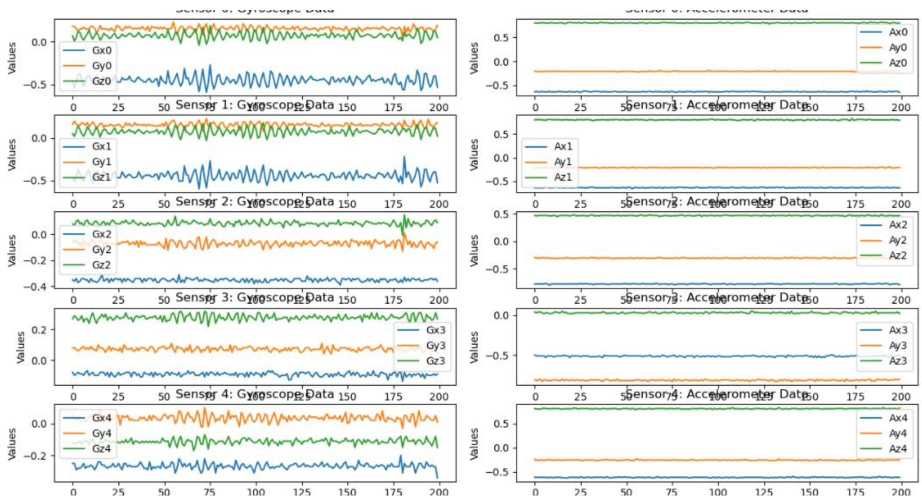
The components necessary for interfacing with the MPU-6050 were selected and prepared to facilitate communication with the sensor and collect the sensor data. A Raspberry Pi was employed as the primary microcontroller to enable communication with the MPU-6050. The sensor was connected to the Raspberry Pi using the I2C communication protocol, with the

SDA (Serial Data) and SCL (Serial Clock) pins of the MPU-6050 connected to the corresponding SDA (GPIO 2) and SCL (GPIO 3) pins on the Raspberry Pi. Furthermore, the VCC and GND pins of the MPU-6050 were connected to the 5V pin and ground pin on the microcontroller.

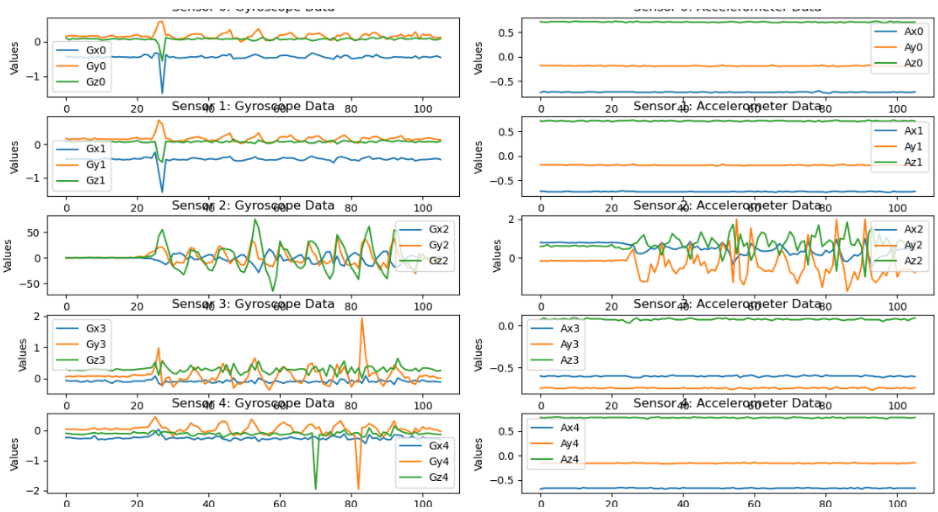
To interact with the MPU-6050 and retrieve the sensor data, the “SMBus" library for Raspberry Pi was utilized, providing functions communicating the sensor to configure it, read the accelerometer and gyroscope data.

The correctness of the data from the MPU-6050 was assessed through several test scenarios and experiments. First, a static test was conducted, wherein the sensor was mounted on a breadboard and kept stationary. The consistency and accuracy of the accelerometer and gyroscope readings were evaluated by comparing these readings to the expected values (0 m/s^2 for the accelerometers, and 0°/s for all gyroscope axes).
Second, a dynamic test was performed, as each sensor was subjected to various motions.

In the static test, the accelerometer readings were found to be consistent. There is little variance in the data. While the gyroscope readings were also found to be consistent with little fluctuation within the range of 0.2 suggesting negligible angular velocity.



In the dynamic test, rapid movements were performed on one of the sensors while the other maintains stationary. It is expected that the one which has significant motion has a high value of acceleration and angular velocity. The result is mostly accurate, while there are one to two noises in the gyroscope data. However, this is acceptable, these exception values can be easily cleaned.



Upon analyzing the data, it was observed that the MPU-6050 demonstrated accurate and consistent performance in both static and dynamic test scenarios. The sensor readings were in line with the expected values, despite minimal noise or discrepancies being detected.

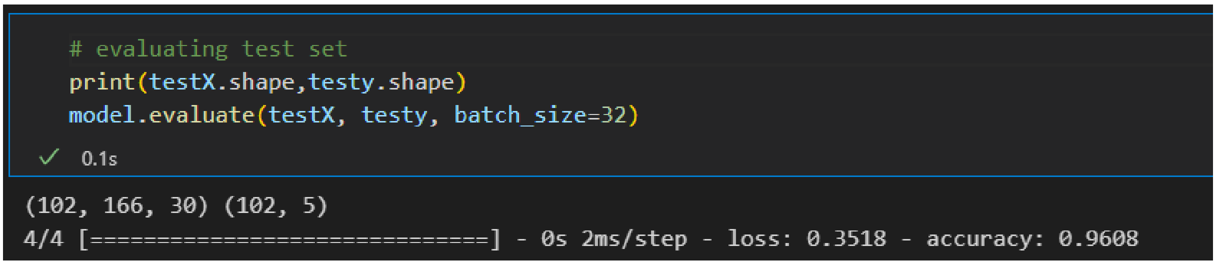
2.3.2 Test the first AI

Past approach: KNN-DTW approach testing

After we created the dataset and developed the first AI, we tested 10 gestures to see whether it is working correctly. We set k to be 3 and recorded 3 gestures of each type. In our result, we were capable of classifying all gestures.

Current approach: Convolutional network testing

In our testing, we separated the data into 80:20 train-test ratio. The accuracy is always above 90% and seldom below 95%, even if there is sometimes overfitting, causing the loss to be relatively higher.



```
# evaluating test set
print(testX.shape,testy.shape)
model.evaluate(testX, testy, batch_size=32)
```

✓ 0.1s

```
(102, 166, 30) (102, 5)
4/4 [==============================] - 0s 2ms/step - loss: 0.3518 - accuracy: 0.9608
```

2.3.3 Test the second AI

After we created the second AI which breaks a sequence into separate gestures, we applied a window size of 4 samples (i.e. $\sim 1/40 \times 4 = 0.1s$). Inside these ranges, we were able to break down "$try\ to\ borrow\ T - shirt$" into "$try$", "$borrow$", "$T - shirt$".
"$Hot\ weather$" into 'Hot', '$weather$'. "$I\ like\ you$" into 'I', '$like$', 'you' in more than 90% accuracy (9 sample files return correct output while 10 are tested).

2.3.4 Test the mobile application and connections

After the application, Bluetooth connection, and the HTTP server have been completed, we applied a series of tests to evaluate the consistency and accuracy of the connection. This includes recording long and short sequences of gestures, testing the time taken to go through the entire process, and the connectivity of the devices under different networks and environments. The results are satisfactory. The connection is stable and responsive. The time taken from the end of performing gestures to a TTS message being played is around 1-1.5 seconds.

2.4 Evaluation

After the testing stage is finished, the system is evaluated based on our goal.

1. Is accuracy acceptable?

   Yes, Both KNN and CNN models achieve superior accuracy (>95%). This is very close to the performance to an state-of-the-art model.

2. Is the system responsive with all those computational loads?

   The KNN failed to respond in real time, since it takes 3 seconds to classify. But after switching to CNN, it only takes less than 0.5 seconds from end-to-end. This make the system very responsive.

3. Can different users use the same system?

   Yes, since the model is trained by data samples from different users. It has learned to cope with noise. So different users with different hand shapes can use the system.

3 Project Planning

3.1 Distribution of Work

Task	**Jimmy**	**Jacky**	**Colin**
Do the literature survey	●	○	○
Design the data reading process	●	○	○
Design the communication between the devices	●	○	○

Design the glove structure and materials	○	○	●
Build the glove prototype with the sensors and Raspberry Pi	●	○	○
Design the algorithm for classifying	○	○	●
Design the basic gesture list	○	○	●
Gather data from the glove	○	○	●
Train the AI model	○	○	●
Design the mobile application UI	○	●	○
Design the mobile application functions and settings	○	●	○
Design the user-defined gesture customization in the app	○	●	○
Develop data processing pipelines	●	○	○
Develop mobile application	○	●	○
Develop user-defined customization	○	●	○
Test the sensors and communication	○	○	●
Test the accuracy of the model	○	○	●
Test the functionality of the mobile application	○	●	○
Write the reports	●	○	○
Prepare for the presentation	○	●	○
Design the project poster	○	○	●

● Leader ○ Assistant

3.2 GANTT Chart

Task	July	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Do the literature survey	▨	▨	▨							
Design the data reading process		▨	▨							

Design the communication between the devices			■	■	■					
Design the glove structure and materials				■	■	■				
Build the glove prototype with the sensors and Raspberry Pi				■	■	■				
Design the algorithm for classifying				■	■	■				
Design the basic gesture list				■	■	■	■			
Gather data from the glove				■	■	■	■			
Train the AI model				■	■	■	■			
Design the mobile application UI				■	■	■	■			
Design the mobile application functions and settings				■	■	■	■			
Design the user-defined gesture customization in the app					■	■	■			
Develop data processing pipelines					■	■	■	■		
Develop mobile application					■	■	■	■		
Develop user-defined customization					■	■	■	■	■	■
Test the sensors and communication						■	■	■	■	■
Test the accuracy of the model						■	■	■	■	■
Test the functionality of the mobile application						■	■	■	■	■
Write the proposal		■	■							
Write the monthly reports				■	■		■			
Write the progress report							■	■		
Write the final report										■
Prepare for the presentation										■
Design the project poster										■

4 Required Hardware & Software

4.1 Hardware

Sensors: 5 boards with MPU6050 and ADC

Microprocessor: Raspberry Pi zero W

Storage: Micro-SD card for RSP OS

Others: breadboard, wire, gloves, mini-HDMI cable

4.2 Software

Flutter/ Android Studio	Mobile app
TensorFlow	AI implmentation

5 References

[1] Trigueiros, Paulo & Ribeiro, Fernando & Reis, Luís. Vision-Based Portuguese Sign Language Recognition System. Advances in Intelligent Systems and Computing. 275. 10.1007/978-3-319-05951-8_57. 2014

[2] T. Chouhan, A. Panse, A. K. Voona and S. M. Sameer, "Smart glove with gesture recognition ability for the hearing and speech impaired," 2014 IEEE Global Humanitarian Technology Conference - South Asia Satellite (GHTC-SAS), 2014, pp. 105-110, doi: 10.1109/GHTC-SAS.2014.6967567.

[3] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and EMG Sensors," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[4] G. Welch, G. Bishop, *An Introduction to the Kalman Filter*, 2006

[5] P. Senin, *Dynamic Time Warping Algorithm Review*, 2008

[6] edufolly (2021). Flutter_bluetooth_serial, https://github.com/edufolly/flutter_bluetooth_serial

[7] Abadi, Mart, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. *Tensorflow: A system for large-scale machine learning. In 12th Symposium on Operating Systems Design and Implementation*

[8] M. Regan, *Timeseries Classification: KNN & DTW*

6 Appendix A: Meeting Minutes

6.1 Minutes of the 1st Project Meeting

Date: May 20, 2022

Time: 2 pm

Place: Zoom

Present: All

Recorder: Lee Wing Piu

1. Discussion items

   1.1 Prof. BENSAOU proposed different topics that we could work on. We chose to work on a project that uses sensors to make a glove to facilitate the hearing-impaired community.

   1.2 Prof. BENSAOU discussed which hardware we should use such as microprocessors and sensors.

   1.3 Prof. BENSAOU explained which fields we should research on such as the machine learning algorithm to classify different hand gestures.

   1.4 Prof. BENSAOU explained the potential difficulties we may face such as different hand shape of people.

2. Goals for the coming week

   2.1 Get all the required hardware to start experimenting with how things work.

   2.2 Distribute the workload to all members evenly so all people have their own parts to do and report their own progress and findings with other groupmates.

6.2 Minutes of the 2nd Project Meeting

Date: July 7, 2022

Time: 3 pm

Place: Learning Commons

Present: Lee Wing Piu (Jimmy), Kwong Chun Yiu (Colin), Sze Tak Ho (Jacky)

Absent: Prof. BENSAOU

Recorder: Kwong Chun Yiu

1. Approval of minutes

   The minutes of the last meeting were approved without amendment.

2. Report on progress

   2.1 All members have begun to read similar projects and existing papers for inspiration and approach for the method and AI models.

   2.2 All members have begun to look for development tools.

   2.3 Jimmy tried to use Raspberry Pi to gather sensor values.

3. Discussion items

   3.1 Jimmy suggested that a mobile app can act as an interface for the users to interact with.

   3.2 We all agreed that the machine learning part will be the most challenging.

3.3 All team members will be having internships this summer, so we won’t be able to do much or meet face to face, but we can still do some work and meet on Zoom/Discord.

4. Goals for the coming month

   4.1 All members will keep doing literature reviews.

5. Meeting adjournment and next meeting

   The meeting was adjourned at 3.30 pm.

   The next meeting will be at 20:30 on August 21st via Discord.

6.3 Minutes of the 3rd Project Meeting

Date: August 21, 2022

Time: 20:30

Place: Discord

Present: Lee Wing Piu (Jimmy), Kwong Chun Yiu (Colin), Sze Tak Ho (Jacky)

Absent: Prof. BENSAOU

Recorder: Sze Tak Ho

1. Approval of minutes

   The minutes of the last meeting were approved without amendment.

2. Report on progress

   2.1 All members have begun to read similar projects and existing papers for inspiration and approaches for the method and AI models.

2.2 All members have reported progress respectively. Jimmy and Jacky have recorded tests of the equipment and the app development using Flutter. Colin has reported findings on “logistic regression”.

2.3 We have been reading different AI algorithms to decide the main approach to take. Colin did research on logistic regression and reported that it may not be optimal for continuously or simultaneously recognizing hand gestures.

3. Discussion items

3.1 Jacky reported that quite a number of other similar projects used neural networks as the main algorithm for recognizing gestures. However, they all lack the ability to recognize continuous movements and more complex gestures, whether it is gloves or cameras. Jacky has also suggested that there could be more functions on the mobile application.

3.2 After the reports from Colin and Jacky about the potential difficulties of recognizing continuous gestures and the breaks between each gesture, Jimmy has suggested an alternative idea that users can use the app to record a period of gestures and label it. The gesture will be recognized next time and also be recorded as another set of data. The goal is to continuously record more deviations of the gestures and improve accuracy. In this approach, the algorithm does not need to recognize every single word and the breaks between them. It can adapt to different users’ habits and adds an important function to the mobile application.

3.3 Jacky and Colin agreed that Jimmy’s suggestion is interesting and would take it as a potential direction to take. All members agreed that we should contact professor BENSAOU for recommendations on our ideas and findings.

3.4 Jimmy contacted professor BENSAOU for a meeting next week and all members are going to start writing the proposal together.

4. Goals for the coming month

4.1 All members will continue reading more papers on different models and algorithms.

4.2 All members will begin drafting the proposal and finalizing the ideas.

5. Meeting adjournment and next meeting

   The meeting was adjourned at -pm.

   The next meeting will be at - on - via Zoom.

6.4 Minutes of the 4th Project Meeting

Date: November 23, 2022

Time: 13:00

Place: Room 3537

Present: All

Recorder: Sze Tak Ho

1. Approval of minutes

   The minutes of the last meeting were approved without amendment.

2. Report on progress

   1. A fully functioning glove prototype with sensors was built.
   2. Several gesture values were extracted and ready to be tested for the Dynamic Time Warping algorithm and KNN.
   3. All members have been researching similar projects and machine learning practices to find the ideal implementation of the AI system.

3. Discussion items

a. Prof. BENSAOU asked questions regarding our design, including the sensors network, glove structure, purpose and usage of the Raspberry Pi and KNN, and reasoning behind choosing the proposed algorithms.

b. Members explained the flow of the project. The Raspberry Pi collects the data from the multiplexing network of 5 sensors on a glove through I2C and simply acts as a transporter. The data will be sent to a mobile device and processed. Jimmy then explained the concept of using KNN as a labeling part after passing the data to the DTW algorithm. Members also expressed they have difficulties figuring out the optimal way to locate the start and end point of a sentence, which is an important problem to be tackled first in the development

c. Prof. BENSAOU recommended members reconsider the choice of KNN since it may not be an efficient algorithm for a project like this. He mentioned that there are many projects on language recognition or translation done with the Long Short-Term Memory algorithm and we could look into LSTM, natural language processing and perhaps reinforced learning as well to solve the efficiency problem.

d. Prof. BENSAOU mentioned our originally proposed method of using a specific gesture to determine the start and end points was not feasible. He indicated that our AI should be capable of determining the start and end points of gestures on its own.

e. Members mentioned that the Chinese University of Hong Kong has a video database of over 8000 sign-language gestures

f. Prof. BENSAOU suggested we could potentially look into transfer learning for our dataset or even ways to link the video to the 6-variable vector.

4. Goals for the coming month

a. All members will continue researching the recommendations from Prof. BENSAOU.

b. All members will begin creating the DTW and classification algorithm.

6.5 Minutes of the 5th Project Meeting

Date: December 28, 2022

Time: 14:00

Place: Room 3537

Present: All

Recorder: Sze Tak Ho

1. Approval of minutes

   The minutes of the last meeting were approved without amendment.

2. Report on progress
   a. After the last meeting, members researched multiple methods to identify the start and end points of gestures, including the sliding window algorithm, frame-by-frame classification, and other AI algorithms.
   b. Members reported difficulties with the transfer learning and video-sensor linking suggestion that Prof. BENSAOU mentioned at the last meeting. There is a lack of related information or similar projects.
3. Discussion items
   a. Jacky reported difficulties in researching the transfer learning and linking method, but members have come up with several promising ideas for detecting the start and end points of sentences.
   b. Prof. BENSAOU responded that it is not necessary to implement transfer learning and other complex techniques in this project. There are better approaches that we can take. Prof. BENSAOU suggested that members should focus on completing the sentence recognition part first before considering other parts that could have simpler solutions.
   c. Prof. BENSAOU mentioned that there are various ways to detect the breaks between each gesture. The simplest way would be doing fixed time slots for each gesture but it is not ideal for practical use. Better approaches include capturing based on the start of

gestures, using LSTM, etc. Prof. BENSAOU recommended members complete the AI part before considering options for making the dataset.

d. Jimmy reported that we could use Google’s Mediapipe which provides 3D coordinates as part of the dataset preparation together with the sign language video database from the Chinese University of Hong Kong.

e. Prof. BENSAOU responded that it is feasible as something that would be nice if we could pull it off. We could utilize the coordinates value and pair it with the sensor values from performing the specific gesture, then feed it into a model to build the database.

4. Goals for the coming month

   a. The start and end point recognition and gesture separation part are expected to be completed within January.

   b. Members will start working on the ideas of preparing the datasets and incorporating the code with the application.

6.6 Minutes of the 6th Project Meeting

Date: February 9, 2023

Time: 15:00

Place: Room 3537

Present: All

Recorder: Sze Tak Ho

1. Approval of minutes

   The minutes of the last meeting were approved without amendment.

2. Report on progress

   a. The Dynamic Time Warping algorithm and KNN model are completed. A basic version of gesture separation is done with the sliding window technique and evaluating the variance of data.

3. Discussion items
   a. Members reported that KNN was still adopted as the classifier instead of other methods suggested in past meetings. The sliding window technique was used for separating the gestures in a sentence. However, the window size currently is a fixed number determined by the test gestures, and it was uncertain what would be the ideal method to find the optimal window size dynamically.
   b. Prof. BENSAOU suggested we look into WaveNet and one of his publications that utilized it with CNN. It would be useful for recognizing patterns and recognition to help the model figure out the best window size.
   c. Members stated that the original plan to break the sequence was silence detection within the gestures by assuming there must be a period where the gesture is held in stasis.
   d. Prof. BENSAOU pointed out that it could work if the result is correct, but there were some problems to be considered such as performing the gestures rapidly, repetition, and forming a meaningful sentence.
   e. Prof. BENSAOU also suggested we contact organizations or the language center to try and find people who know sign language to acquire realistic data, which can help develop the model better. He also suggested we check out dataset sources such as Kaggle to search for suitable datasets if possible.
4. Goals for the coming month
   a. Complete the gesture-breaking part and sentence-forming parts of the AI.
   b. Start to build the dataset by researching databases or recording gestures manually.

6.7 Minutes of the 7th Project Meeting

Date: March 16, 2023

Time: 16:00

Place: Room 3537

Present: All

Recorder: Sze Tak Ho

5. Approval of minutes

   The minutes of the last meeting were approved without amendment.

6. Report on progress
   a. More gestures were recorded to test the sliding window method and the AI. The mobile application is at the last stage of development
7. Discussion items
   a. Members reported that during the development of the Flutter application, the method to incorporate Python scripts of the AI into the application was chosen to be an HTTP request to a server running the AI. Members were not sure about whether it is an ideal or a fair approach.
   b. Prof. BENSAOU asked about the properties of Flutter and thought that a server running on a powerful PC could work.
   c. Members reported that the speed of the KNN is slow. It took 2 to 4 seconds to classify the gestures.
   d. Prof. BENSAOU commented that the speed is unreasonably slow and suggested that other approaches should be adopted. He mentioned that another FYP group has successfully utilized LSTM in a similar manner and suggested that we try to study and

incorporate LSTM. He also mentioned that application development should be a lower priority than AI as the progress was relatively slow.

8. Goals for the coming month
   a. Improve the speed of the AI by changing to other methods such as LSTM
   b. Finish up the remaining part of the project

6.8 Minutes of the 8th Project Meeting

Date: April 20, 2023

Time: 13:30

Place: Room 3537

Present: All

Recorder: Sze Tak Ho

9. Approval of minutes

   The minutes of the last meeting were approved without amendment.

10. Report on progress
    a. The classification algorithm has changed from KNN to CNN. The mobile application, Bluetooth connection, and server setup are all completed.
11. Discussion items
    a. Members reported that instead of LSTM, CNN was adopted as the classification method. It is much faster than KNN with high accuracy with tests. The other parts of the project have been finished and could be further optimized.
    b. Prof. BENSAOU asked about the intention behind the various decisions such as choosing CNN, the sample rate of data, and the window size. He explained that if the reasons are justified, it is fine that the results contain certain flaws or drawbacks. He

recommended that the project should be further worked on and polished before the presentation. Prof. BENSAOU also suggested the MIT deep learning course to get information on LSTM and view examples of other projects.

   c. Members did a small demo of the glove and classification in action. Prof. BENSAOU advised that the software side should be kept more centralized and the method of using a button to control inside the application is not ideal. The dataset was also too small, and part of the preparation was wasted on non-sign-language gestures. Members should continue to expand the dataset, and either work on alternative solutions or fine-tune the current implementations before the presentation.

12. Goals for the coming month
    a. Record more gestures to expand the dataset for classification
    b. Further study LSTM and decide on the AI, tune other aspects of the project