2023 年臺灣國際科學展覽會
優勝作品專輯

作品編號	**190006**
參展科別	電腦科學與資訊工程
作品名稱	**Human-computer Interaction-based Millimeter-wave Radar Gesture Recognition**
得獎獎項	三等獎

就讀學校	臺中市華盛頓高級中學
指導教師	鮑興國
作者姓名	邱堉綸

關鍵詞 **Millimeter-wave Radar、Gesture Recognition、Machine Learning**

作者簡介

我是邱堉綸，就讀華盛頓高級中學二年級，我在國二時開始接觸 Python，便對程式設計領域感到好奇，因此後來在線上學習了 C++語言、資料結構、演算法、資料科學和 Linux 作業系統等，進而參加科教館舉辦的 2022 未來之星智慧科技營隊，且有了這次參加臺灣國際科展的機會，讓我有機會將自己的想法實作並展示出來。過程中特別感謝鮑興國教授及翁子謙學長的指導與幫助，除了教導我許多「機器學習」領域的相關知識，更讓我體會如何以嚴謹的科學態度來進行研究。未來，我也會持續在資訊領域努力，繼續朝夢想前進。

Abstract

This study presents a real-time dynamic gesture recognition technique with millimeter-wave radar, which operates the application by several simple dynamic gestures instead of keyboard and mouse, thus providing a more friendly and intuitive Human-computer Interaction (HCI). We conducted gesture attribute analysis, sensor data representation evaluation, learning model efficiency evaluation, and system live testing performance analysis to improve the usability and maneuverability of the gesture control human-machine interface. Our learning model is based on a hybrid model (1DCNN+LSTM) of size 415 KB for four dynamic gestures and runs gesture recognition at a sampling rate of 30 FPS on a Texas Instruments FMCW radar evaluation board. We achieved 94.5% accuracy for media player application among seven users, including five right-handers and two left-handers. In addition, our approach enables interoperable applications in complex spaces outside the controlled laboratory environment without significant misidentification.

本研究提出了一個毫米波雷達即時動態手勢辨識技術，透過幾個簡單的手勢取代鍵盤和滑鼠來操作應用程序，從而提供更生活化和直覺化的人機介面。我們透過手勢屬性分析、手勢訓練資料格式選擇評估、學習模型效能評估和系統實測性能分析，以提高手勢控制人機界面的實用性。我們的學習模型採用一大小為 415 KB 的 1DCNN+LSTM 混合模型支持四個動態手勢，並在德州儀器的 FMCW 雷達評估板上以 30 FPS 的採樣速度進行手勢識別。我們在 7 個用戶（包括 5 個右撇子和 2 個左撇子）的多媒體撥放實際測試中達到 94.5%的操控準確率。此外，我們的方案在實驗室環境之外的複雜空間中操控應用程序，也不會有明顯的辨識錯誤的情況發生。

1. Introduction

Human-computer Interaction (HCI) has progressed from keyboards, mice, and touch to more diverse and intuitive control methods, while gesture interactions have been studied in the HCI community by many researchers. Millimeter-wave radar sensor uses radio waves, which is not interfered by sound and light sources, to achieve excellent sensing capabilities to measure the distance, speed and direction of moving objects. With advancements in machine learning, incorporating gesture recognition and gesture tracking technology with millimeter-wave frequency modulated continuous wave (FMCW) radar has extremely important application value in the HCI community.

We consider the following design requirements to design a gestural interaction system for selected applications, making the application easier to use in all environments.

- Usability

How easy is it to perform the gesture. Users feel natural and easy to pick up on with a short learning curve.

- Accessibility

Ensure users can interact with devices easily without any difficulty.

- Comfortability

Consider the human body ergonomics, if a gesture is uncomfortable or too repetitive, the experience will not be great for users.

- Intuitiveness

Avoid using complex gestures which are hard to learn, instead, choose gestures that allow users understand them instinctively.

Through millimeter-wave radar detection and real-time dynamic gesture recognition technology, a more user-friendly and intuitive human-machine interface is provided, thereby making the communication between humans and machines more barrier-free. Our main contributions are summarized as follows:

- A qualitative analysis approach based on quantity, function support, complexity and attributes is proposed to design a user-friendly gesture set to interact with the selected application.
- Discussing strategies for selecting sampling rates and sensor data representations during the data acquisition phase to improve model efficiency and performance.
- Discussing strategies for model layer construction and sliding window selection to achieve high recognition accuracy and performance.
- Real-time and power-saving implementation on a 60GHz FWCW radar solution to evaluate the proposed gestural interaction system efficiency and readiness.
- Measuring the reliability of millimeter-wave radar sensors to external environmental interferences, including four different types of interference: Bluetooth, electromagnetic waves (hair dryer), fan swiping and millimeter-wave radar.

The rest of this paper is organized as follows. In Section 2, we present the study of gesture recognition technology in the HCI community. In Section 3, we demonstrate an efficient millimeter-wave radar dynamic gesture control system for the selected application. The design flow and its performance evaluation are described in Section 4. The interaction with the selected application snapshots is presented in Section 5. The conclusions of this study are described in Section 6.

2. Related Work

Existing gesture interaction methods use image sensors [1, 2, 3] to capture the gesture movement and there is significant breakthrough among them. However, image sensor suffers from several difficulties such as sensitive to light and atmospheric conditions, which resulting in insufficient sensing capabilities. By overcoming many of the problems with camera-based approaches, millimeter-wave radar has the potential to become the basis for gesture recognition. Most gesture-sensing work uses millimeter-wave FMCW radar, builds training datasets using spectrograms of range-Doppler images generated from raw data, and captures features from images using machine learning models capable of recognizing gestures [4, 5, 6]. Furthermore, since dynamic gestures are time-series data, a sliding-window preprocessing layer needs to be added to the learning model to trade off accuracy and inference time [7]. A qualitative and quantitative analysis of novel radar-based recognition solutions relevant to HCI applications are reviewed in [8].

In this paper, we used Cartesian coordinates instead of the spectrogram of the range Doppler image as the inference basis for the learning model. Our learning model performs gesture detection and classification based on a hybrid model consisting of four 1D-CNNs and one LSTM network. Furthermore, we evaluated model performance through live system testing on real-world applications in complex spaces outside the controlled laboratory environment.

3. Proposed Method

Gestures are quite primitive and natural expressions in daily life. It has become a trend for wearables to incorporate dynamic gestures to make them easier to use. Dynamic gestures are the changes of gestures in continuous time, including the changes in gesture shape and trajectory. Using FMCW technology to detect the position, speed and direction of moving objects ahead through millimeter-wave is the most popular short-range gesture recognition method today.

3.1 Overview of gesture interaction system

The FMCW radar gesture recognition system first converts the complex dynamic gesture information from three-dimensional space into one-dimensional data points to reduce the computing time, and calculates the features of each gesture through a learning algorithm, which is used as a recognition basis to identify different gestures to achieve real-time human-computer interaction effect.

3.2 Design flow and system performance metrics

First, we proposed a systematic design flow for the selected applications, including gesture design (number of gestures, gesture attribute analysis and the function mappings between gestures and the application, etc.) and model design (gesture recognition rate, inference time, model size and window size, etc.), as shown in Figure 1. Next, we deployed the model with best performance on the IWR6843AOPEVM board for system live testing efficiency evaluation. Then, users with different dominant hand tested the board to measure several performances: (1) Is the interaction between the gestures and application conforms to human natural habits? (2) Is the gesture easy to perform and remember? (3) How fast is the response time? Finally, according to the evaluation results of the system, we refined the gesture and the model iteratively to achieve better performance.

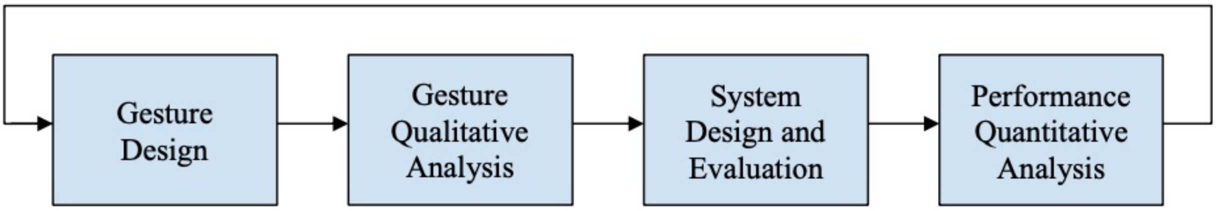

Gesture Design
Gesture Qualitative Analysis
System Design and Evaluation
Performance Quantitative Analysis

Figure 1: Dynamic gesture control system design flow.

3.3 Toolchains and hardware setup

This study uses TI IWR6843AOPEVM (Table 1), and the relevant hardware parameter settings are shown in Table 2. The software tools and firmware versions used are listed in Table 3.

Table 1: TI IWR6843AOPEVM specifications.

Number of transmit antennas	3
Number of receiving antennas	4

Transmit power	12 dbm
Frequency	60 ~ 64 GHz
Peripheral communication interface	I2C、LVDS、QSPI、SPI、UART

Table 2: TI IWR6843AOPEVM millimeter wave radar settings.

Number of transmit antennas	3
Number of receiving antennas	4
Frequency	60 GHz

Table 3: Software tools and TI IWR6843AOPEVM firmware versions.

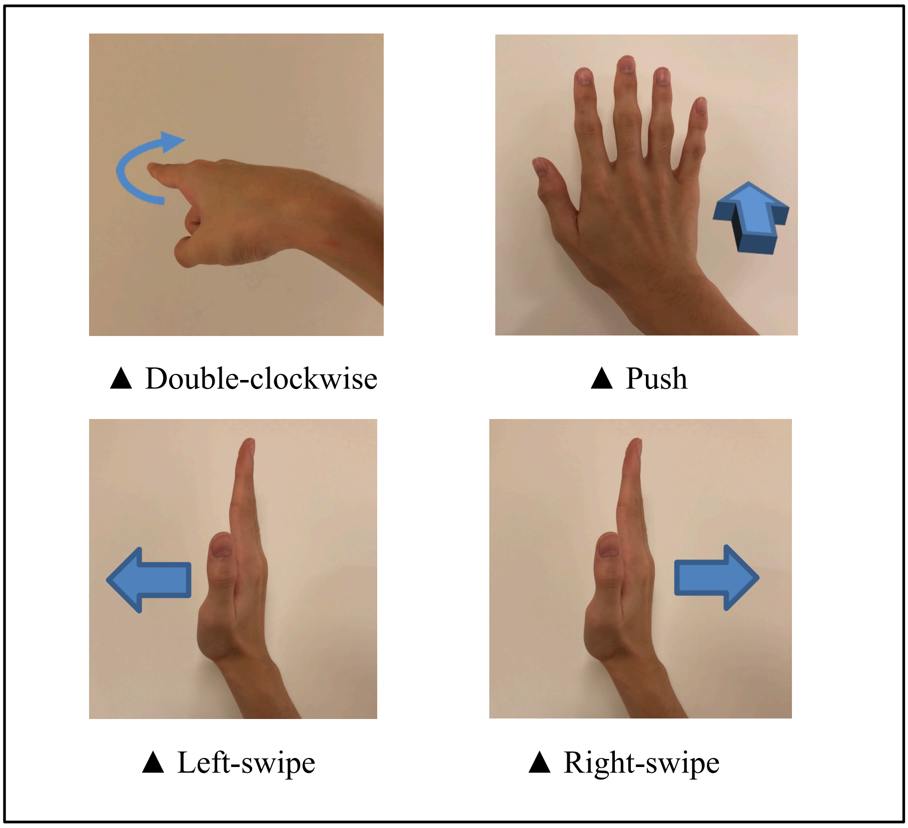
Imagimob Studio	Version 3.3.480
Python	Version 3.6.7
AutoIt	Version 3.3.16.0
IWR6843AOPEVM Firmware SDK	16.9.6.LTS

4. Finding

Based on the user experience design principles, this study divides the design flow of the gestural interaction system into six stages: gesture design, gesture attribute analysis, data collection and pre-processing, model design and efficiency evaluation, system design, and system performance metrics and analysis.

4.1 Gesture design

Taking a gesture-controlled multimedia player application as an example, the control functions are divided into three categories, play/pause, track control, and volume control. Here we design two sets of gesture A and B. Group A has three gestures, one-tap, double-tap and triple-tap. Group B (Figure 2) has four gestures, double-clockwise, right-swipe, left-swipe and push. Based on the simplified gestures for common interactions mechanism, the gestures and control function mappings of Group A and Group B are shown in Table 4.



▲ Double-clockwise

▲ Push

▲ Left-swipe

▲ Right-swipe

Figure 2: Group B gestures.

Table 4: Mapping table between gestures and multimedia player functions.

	Gesture	Function support	Complexity (Duration, ms)
Group A	One-tap	Play / Pause	Low (825)
	Double-tap	Next track	Middle (970)
	Triple-tap	Previous track	High (1683)
Group B	Double-clockwise	Mode switch	Middle (1223)
	Right-swipe	Next track / Volume increase	Low (523)
	Left-swipe	Previous track / Volume decrease	Low (775)
	Push	Play / Pause	Low (675)

4.2 Gesture attribute analysis

Gesture design should strive to be intuitive, easy to use, and ergonomic. More complex and excessive gestures, in addition to increasing the complexity and computing time of the machine learning model, not only make it difficult for users to get started, but also reduce usability. This study proposes four gesture attributes to evaluate the strengths and weaknesses of the gesture sets, as shown below.

- Uniqueness

The similarity between gestures is low. Taking one-tap and double-tap as an example, double-tap is unique in that a one-tap is not mistaken for a double-tap.

- Logical

The mapping between gestures and functions is intuitive and easy to remember. For example, the functions of previous track and next track are logically opposite, and their mapping gestures also need to have opposite semantics like right-swipe and left-swipe.

- Calibration

Gestures need to be pre-calibrated. For example, the long and short swiping in the same direction are different for radar detection, and this must be taken into account during data collection phase to capture data at different speeds.

- Prefix code

A gesture is a subset of another gesture. For example, one-tap is the prefix code of double-tap, which increase the likelihood that double-tap will be recognized as one-tap twice.

Figure 3 shows the Cartesian waveforms and durations of Group A gestures. Gesture durations in Group A range from 0.5s to 1.7s, with triple-tap having the longest duration. The y-axis (yellow line) waveform of one-tap shows one peak, the y-axis (yellow line) waveform of double-tap shows two peaks, and the y-axis (yellow line) waveform of triple-tap shows three peaks. From the waveform point of view, there is also a prefix code relationship between this group of gestures, which increases the possibility that double-tap is recognized as one-tap and triple-tap is recognized as double-tap. Although this group of gestures is simple, it does not have any logicality and semantics, and users need to spend more time remembering the mapping between gestures and functions.

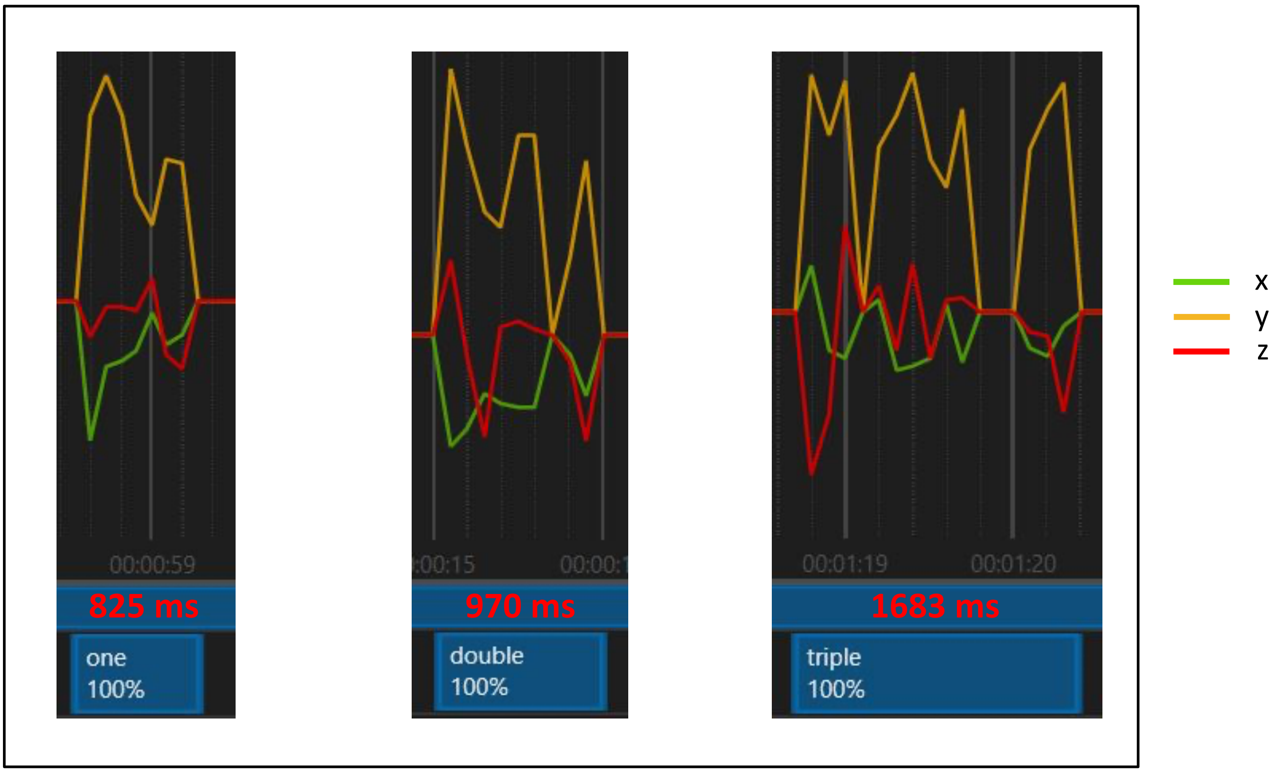


Figure 3: Cartesian waveforms and gesture durations of Group A. There is a prefix-code relationship between this group of gestures, which increases the possibility of misidentifications.

Figure 4 shows the Cartesian waveforms and durations of Group B gestures. Gesture durations in Group B range from 0.4s to 1.5s, with double-clockwise having the longest duration. Right-swipe and left-swipe have their x-axis (green line) as the main feature, push has their z-axis (red line) as their main feature. In double-clockwise, the x, y, and z axes are all changed, and is more unique to distinguish from others. In addition, right-swipe and left-swipe in this group have opposite meanings, and it is not only intuitive but also easy to remember to use them to do track control and volume control. However, here we use millimeter-wave to detect gestures, both right-swipe and left-swipe require more data captured by different dominant hands at different speeds to make the model more robust.

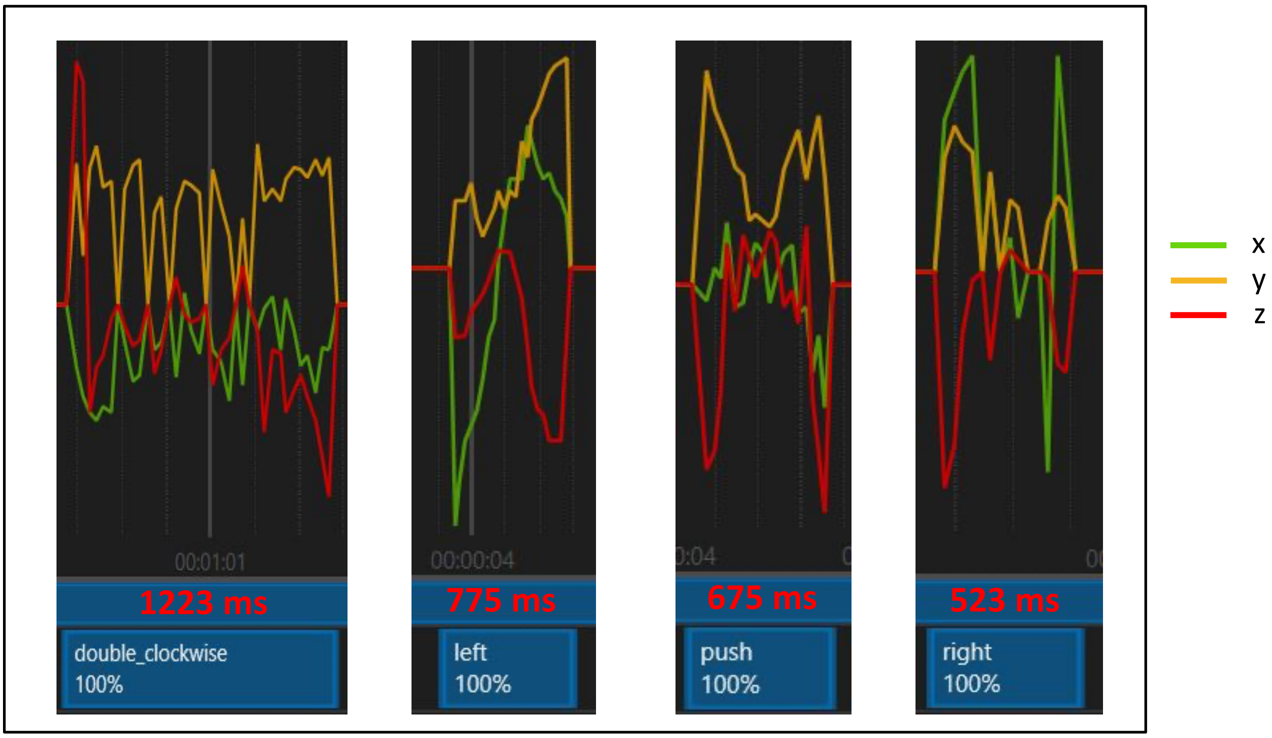


Figure 4: Cartesian waveforms and gesture durations of Group B. Swiping left and right are characterized by the x-axis, while push is characterized by the z-axis. Also, swiping left and right have opposite meanings, making them intuitive for track and volume control.

Table 5 shows the qualitative analyzing results of Groups A and B based on the above proposed gesture attributes. From the results, From the results, there is a prefix code relationship in group A, which increases the possibility of misidentification. Group B has good logic, and there is no subset relationship between gestures, which is better than group A. We will use Group B to control the media player.

Table 5: Qualitative analysis results of gesture attributes for Group A and Group B.

	Gesture	Prefix code	Calibration	Uniqueness	Logical
Group A	One-tap	No	No	No	No
	Double-tap	Yes	No	No	No
	Triple-tap	Yes	No	Yes	No
Group B	Double-clockwise	No	No	Yes	No
	Right-swipe	No	Yes	No	Yes
	Left-swipe	No	Yes	No	Yes
	Push	No	No	No	No

4.3 Gesture data collection and pre-processing

Data collection is one of the most time-consuming processes in machine learning stages. We used the Imagimob studio tool to capture the radar signals of TI IWR6843AOPEVM as training data. The toolchain provides four kinds of captured information: Cartesian coordinate, polar coordinate, velocity and noise, where Cartesian and polar coordinates are the positions of moving objects, expressed in different coordinate systems under the same origin, velocity is the speed of the moving object, and noise is the interference data. In addition, the detection range is centered on the radar, with a plane of 30cm and a depth of 30cm. It can detect up to four moving objects and offers the closest and fastest options. In data collection phase, we need to consider the sensor data format, sampling rate, and label strategy.

- Data representation

In the beginning, we used four data types and four closest objects, but the motion trajectories of the 4 closest objects are similar, as shown in Figure 5, one closest object data point is sufficient for model training. Figure 6 is the waveform of the push gesture captured at different speeds. The velocity waveform is in a state of high variation, and adding velocity data limits the user to test the system at the same speed during the data collection phase. In Figure 7, the range data (blue line) in polar coordinate is almost unchanged, one less feature than in Cartesian coordinate. Table 6 shows that adding noise does not improve F1 score or miss rate. Based on the results of the above analysis and discussion, we chose the Cartesian coordinates of the closest object as the features of our training data.

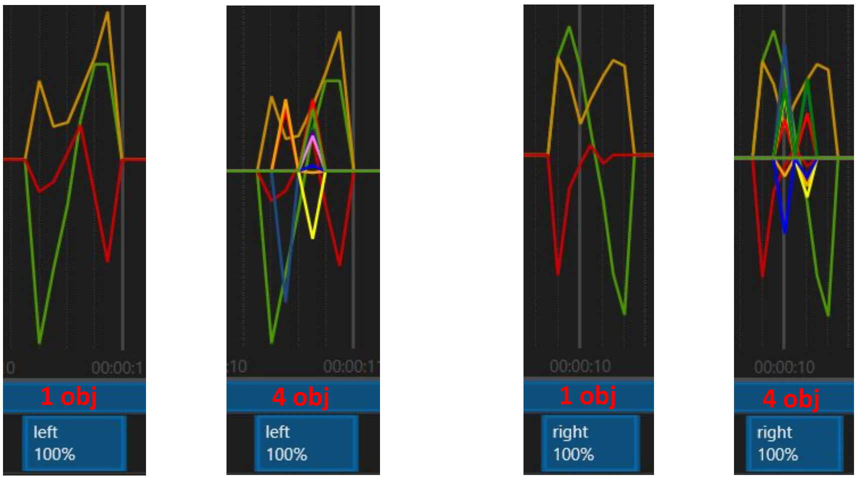


Figure 5: Cartesian coordinate waveform of left-swipe and right-swipe, the left side is one object and the right side is four objects. The finger movements for left-swipe and right-swipe are similar.

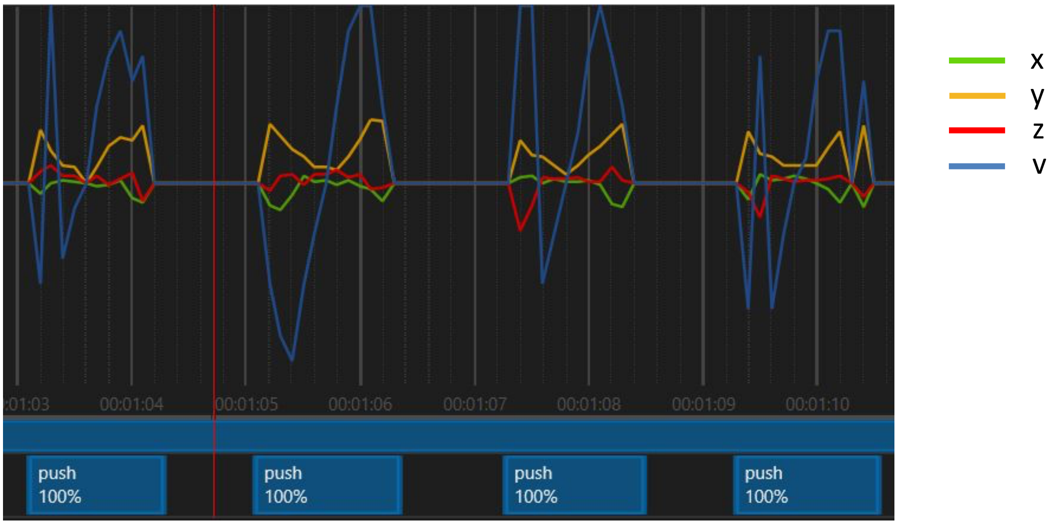


Figure 6: Cartesian coordinate and velocity (blue line) waveform of push gesture at different speeds. Velocity waveform is not a key feature.

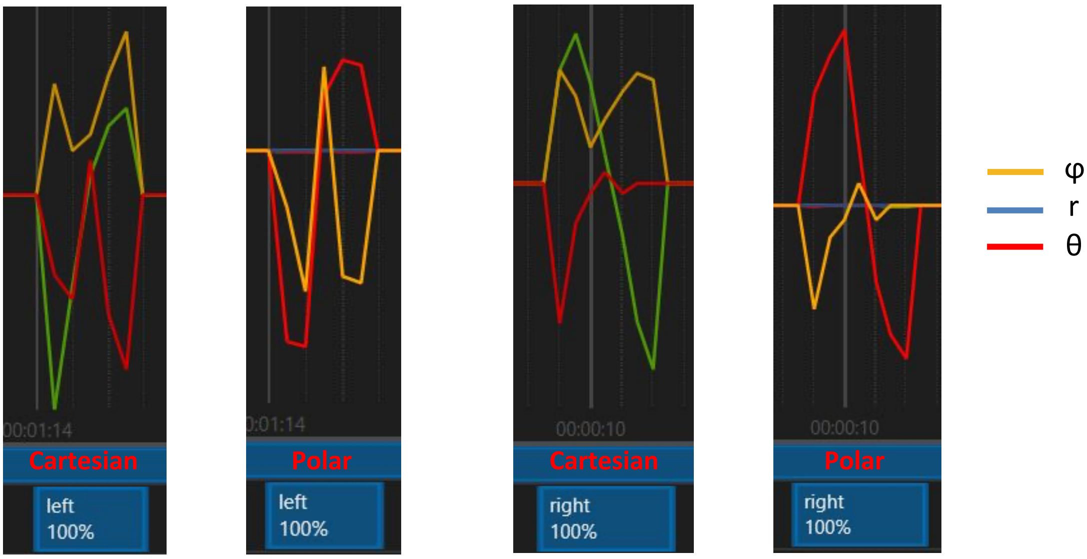


Figure 7: Gesture waveform of left-swipe and right-swipe, the left side is Cartesian coordinate and the right side is polar coordinate. The range data (blue line) in polar coordinate shows little change.

Table 6: F1-score and miss rate comparisons between Cartesian and Cartesian + noise. Adding noise does not improve F1 score or miss rate.

Data type	**Cartesian**		**Cartesian + Noise**	
	F1 Score	Miss Rate	F1 Score	Miss Rate
Double Clockwise	97.04%	1.88%	97.93%	1.98%
Left	96.20%	3.63%	94.08%	4.49%
Push	93.62%	2.81%	95.27%	2.98%
Right	96.28%	3.80%	96.98%	2.01%

- Sampling rate

In our first attempt, we captured data at a sampling rate of 10 FPS, but some important features are lost if the gesture moves too fast, as shown in Figure 8. Furthermore, by comparing the F1 scores for 10 FPS and 30 FPS sampling rates, high data sampling rates can improve model performance by 6.7%, especially for left-swipe and right-swipe, as shown in Figure 9. Here, all training datasets are captured at a sampling rate of 30 FPS by three testers at slightly different speeds.

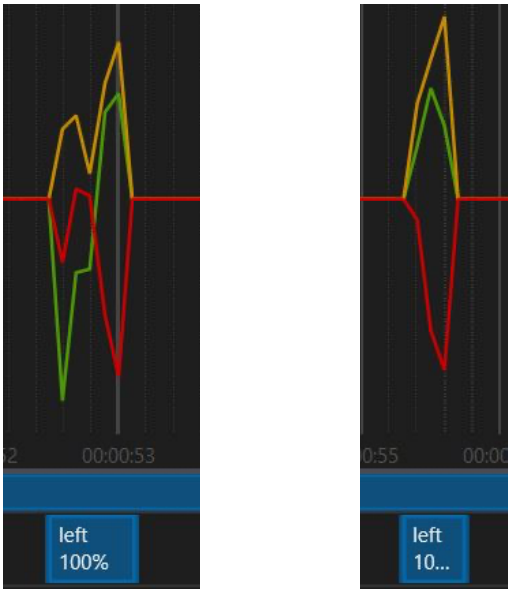


Figure 8: The waveform of left-swipe at 10 FPS sampling rate. The left side is normal and the right side shows that some key features are lost at high-speed hand motion.

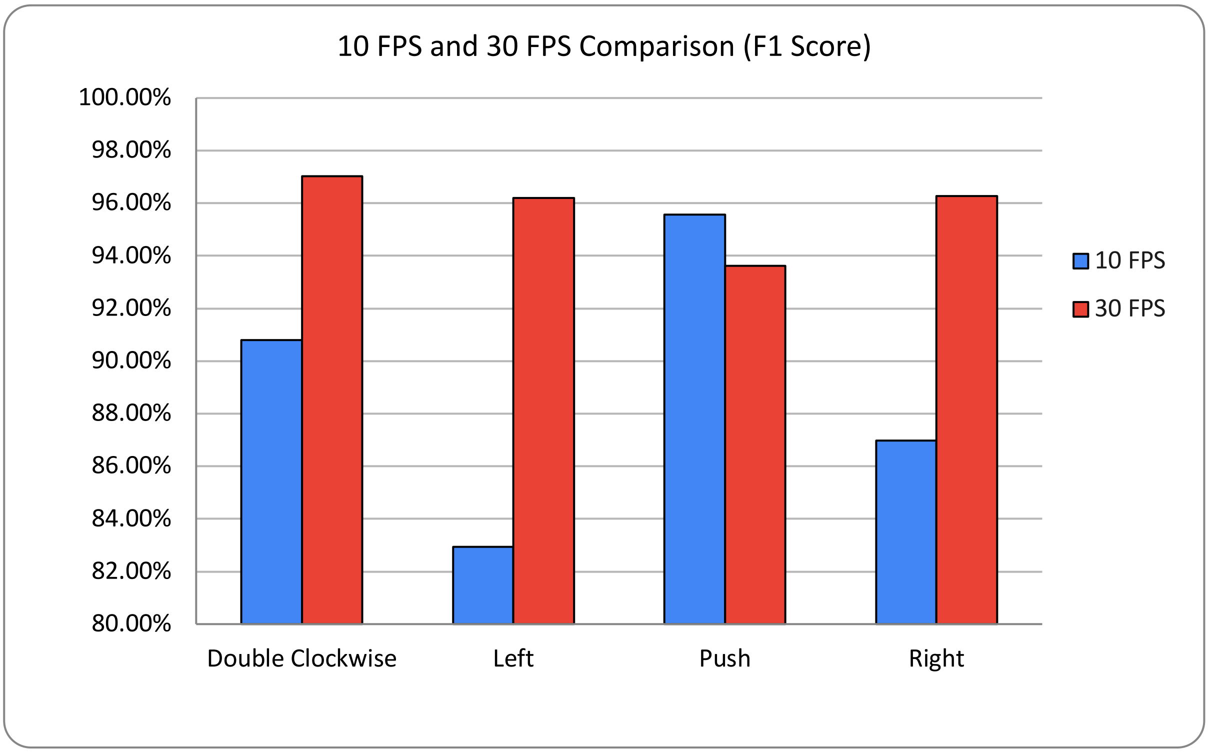


Figure 9: F1-score comparisons between 10 FPS and 30 FPS sampling rate. 6.7% improvement at 30 FPS, especially for left-swipe and right-swipe.

- Data labeling

At the end of the data collection phase, we annotate a label to each waveform with a gesture category in the data track. With the help of video, the captured data points are labeled as gesture categories by human inspection for each data track, as shown in Figure 10. Unlabeled data points in the data track are treated as unknown gestures. Also, if there is an ambiguous waveform in the data track, this data track will be discarded.

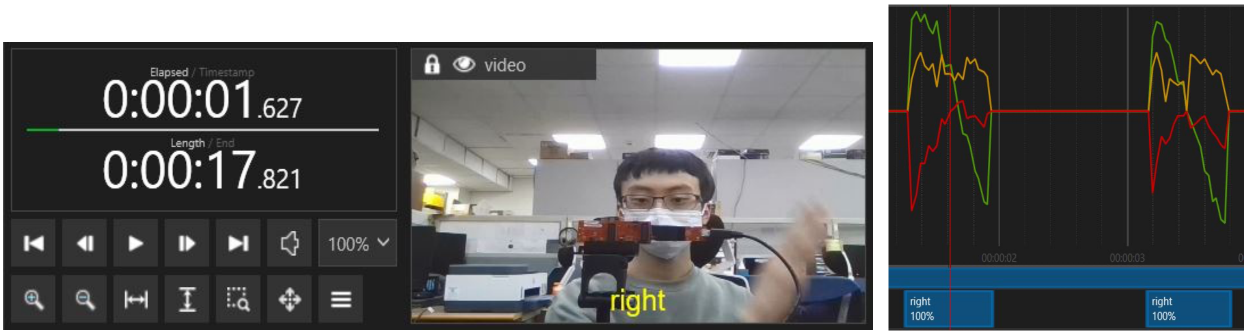


Figure 10: Data labeling, the left is the video track and the right is the waveform track. Labels are annotated by human inspection with the aid of video to avoid mislabeling.

4.4 Model design

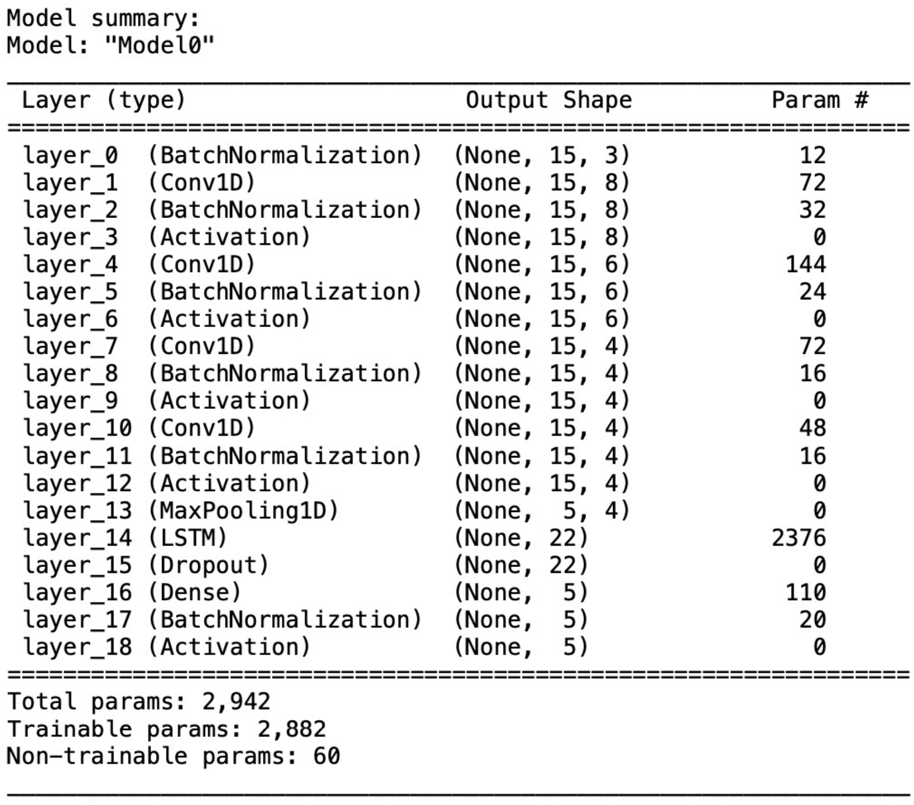
Since dynamic gestures are a series of motion data points, we use a hybrid model consisting of Convolutional Neural Network (CNN) layers and Long Short-Term Memory (LSTM) layers. CNN helps to extract the features of the gesture data, while LSTM helps to memorize the long sequence of the data. The model design is divided into the following three steps:

- Model building

According to the gesture data points of Group B, Imagimob tool provides multiple simple models based on CNN, and then determines the best model structure by comparing the F1-score performance of each gesture under different models. However, due to the time-series data of dynamic gestures, we added an LSTM layer and turned on its bi-direction option for comparison. As shown in Table 7, CNN-LSTM performs best in terms of F1 score and miss rate compared to CNN and CNN-BiLSTM. We choose CNN-LSTM model for the final system performance metrics and evaluation. Figure 11 shows the number of layers of the final CNN-LSTM model and the parameters of each layer.

Table 7: F1 score and miss rate comparisons among CNN, CNN-LSTM and CNN-BiLSTM. CNN-LSTM performs best in terms of F1 score and miss rate.

Model	**CNN**		**CNN-LSTM**		**CNN-BiLSTM**	
	F1 Score	Miss Rate	F1 Score	Miss Rate	F1 Score	Miss Rate
Double Clockwise	96.03%	2.74%	97.04%	1.88%	96.23%	1.88%
Left	96.39%	4.72%	96.20%	3.63%	96.26%	4.96%
Push	92.56%	7.63%	93.62%	2.81%	94.19%	2.41%
Right	96.37%	3.63%	96.28%	3.80%	95.96%	3.63%



```
Model summary:
Model: "Model0"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 layer_0  (BatchNormalization)  (None, 15, 3)          12
 layer_1  (Conv1D)              (None, 15, 8)          72
 layer_2  (BatchNormalization)  (None, 15, 8)          32
 layer_3  (Activation)          (None, 15, 8)           0
 layer_4  (Conv1D)              (None, 15, 6)         144
 layer_5  (BatchNormalization)  (None, 15, 6)          24
 layer_6  (Activation)          (None, 15, 6)           0
 layer_7  (Conv1D)              (None, 15, 4)          72
 layer_8  (BatchNormalization)  (None, 15, 4)          16
 layer_9  (Activation)          (None, 15, 4)           0
 layer_10 (Conv1D)              (None, 15, 4)          48
 layer_11 (BatchNormalization)  (None, 15, 4)          16
 layer_12 (Activation)          (None, 15, 4)           0
 layer_13 (MaxPooling1D)        (None,  5, 4)           0
 layer_14 (LSTM)                (None, 22)           2376
 layer_15 (Dropout)             (None, 22)              0
 layer_16 (Dense)               (None,  5)            110
 layer_17 (BatchNormalization)  (None,  5)             20
 layer_18 (Activation)          (None,  5)              0
=================================================================
Total params: 2,942
Trainable params: 2,882
Non-trainable params: 60
_________________________________________________________________
```

Figure 11: A hybrid model consisting of four 1D-CNNs and one LSTM network for gesture recognition. Since the gesture durations in group B range from 0.4s to 1.5s, a multi 1D-CNN is used to filter key features.

- Sliding window

Since the gesture durations in Group B range from 0.4s to 1.5s, a sliding window preprocessing layer needs to be added to trade off accuracy and inference time. We measured both F1 score and miss rate of CNN-LSTM model under different window sizes to determine the most suitable window size. According to the comparison results in Table 8, the most appropriate window size is 15 in terms of F1 score and miss rate. In addition, the inference time of window size 15 is less than 2ms during system live test.

Table 8: F1 score (left) and miss rate (right) comparisons between different window sizes. A window size of 15 (0.5s) is in terms of F1 score and miss rate, and also accommodates the shortest gesture duration.

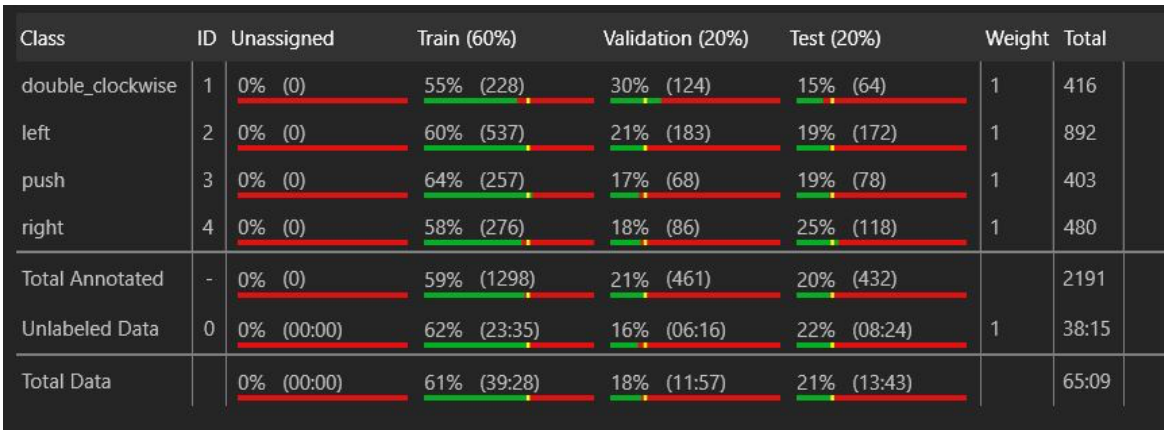
Window size	**10**		**15**		**20**	
	F1 Score	Miss Rate	F1 Score	Miss Rate	F1 Score	Miss Rate
Double Clockwise	96.12%	1.27%	97.04%	1.88%	96.15%	1.96%
Left	96.51%	4.89%	96.20%	3.63%	93.35%	4.62%
Push	96.09%	2.41%	93.62%	2.81%	80.68%	22.83%
Right	96.81%	4.62%	96.28%	3.80%	92.34%	9.21%

- Dataset distribution

We trained the model using the distribution ratios (60%, 20%, 20%) of the training, validation, and test sets, as shown in Table 9, and the dataset is automatically distributed. Furthermore, after finding the best model, we fixed the dataset distribution for all other performance evaluations to achieve a fair result.

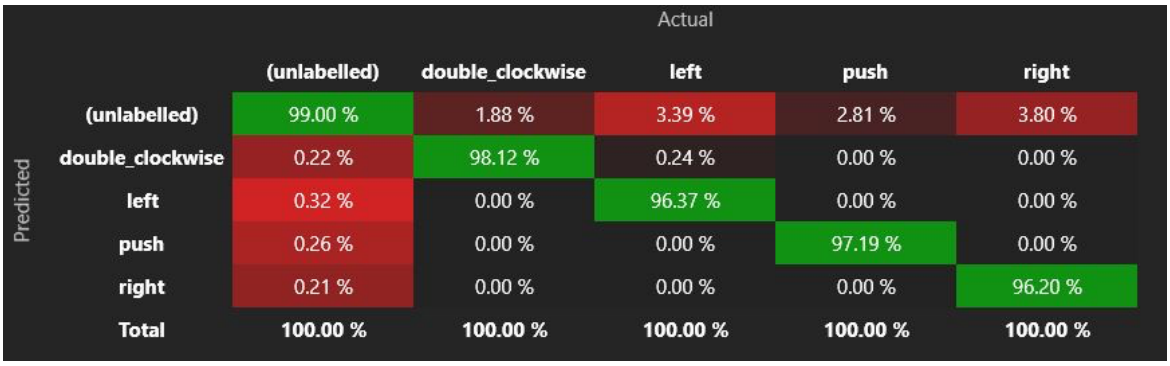
Table 10 is the confusion matrix of the final CNN-LSTM model, the accuracy of each gesture can achieve higher than 96%. According to the convergence plot in Figure 12, 2191 pieces of data are enough to train the model without underfitting. In addition, the validation loss is greater than or equal to the training loss during the training process and there is no overfitting situation.

Table 9: Redistribute training, validation, and test datasets by 60, 20, 20 for model performance optimization. Swipe left and right are mainly characterized by the x-axis, requiring more datasets to improve accuracy.



Class	ID	Unassigned	Train (60%)	Validation (20%)	Test (20%)	Weight	Total
double_clockwise	1	0% (0)	55% (228)	30% (124)	15% (64)	1	416
left	2	0% (0)	60% (537)	21% (183)	19% (172)	1	892
push	3	0% (0)	64% (257)	17% (68)	19% (78)	1	403
right	4	0% (0)	58% (276)	18% (86)	25% (118)	1	480
Total Annotated	-	0% (0)	59% (1298)	21% (461)	20% (432)		2191
Unlabeled Data	0	0% (00:00)	62% (23:35)	16% (06:16)	22% (08:24)	1	38:15
Total Data		0% (00:00)	61% (39:28)	18% (11:57)	21% (13:43)		65:09

Table 10: Confusion matrix for CNN-LSTM model. Each gesture is more than 96% accurate.



		Actual				
		(unlabelled)	double_clockwise	left	push	right
Predicted	(unlabelled)	99.00 %	1.88 %	3.39 %	2.81 %	3.80 %
	double_clockwise	0.22 %	98.12 %	0.24 %	0.00 %	0.00 %
	left	0.32 %	0.00 %	96.37 %	0.00 %	0.00 %
	push	0.26 %	0.00 %	0.00 %	97.19 %	0.00 %
	right	0.21 %	0.00 %	0.00 %	0.00 %	96.20 %
	Total	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %

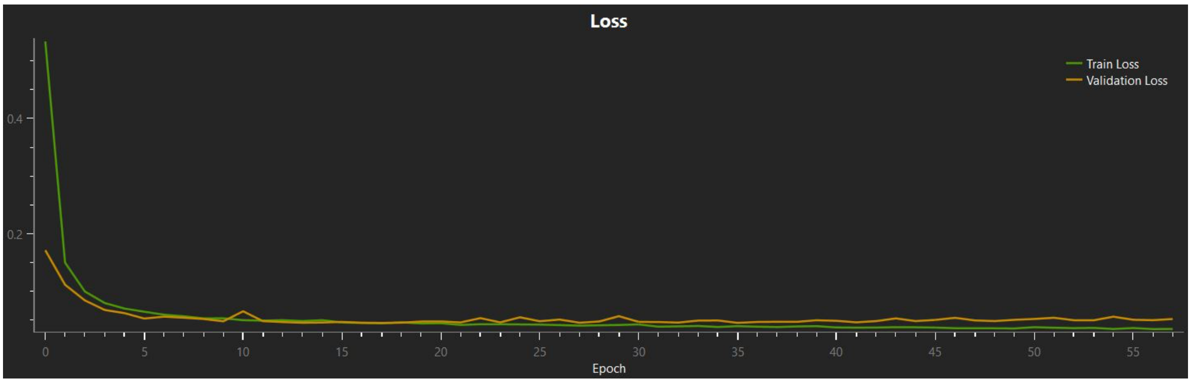


Figure 12: Convergence plot for CNN-LSTM model.

4.5 System design

The prototype of the proposed gesture recognition system is shown in Figure 13. First, we deployed the model on the IWR6843AOPEVM board through the TI UniFlash tool. The millimeter-wave radar sensor device will output its gesture recognition results to the teraterm terminal through UART interface, then use the python script to parse the UART output to obtain the gesture ID. Finally, run the AutoIt script to operate the mapping function according to the gesture ID.

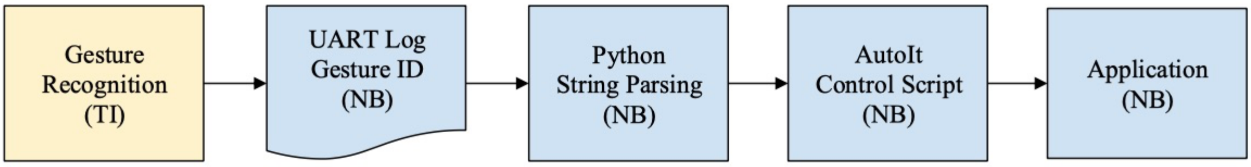


Figure 13: The proposed gesture recognition system prototype.

Figure 14 is the firmware flow chart. The device is initially in sleep power saving mode. To enter wake-up mode, simply perform a random gesture within the detection range. Next, the model will calculate the current window data every 0.75s and perform gesture prediction, and finally output the recognition results to the teraterm terminal through the UART interface. In addition, if no gesture appears within the detection range for more than 10s, the device will enter sleep mode again. Furthermore, we found that the delay gap between gesture detection and gesture prediction must be set to be greater than 0.75s, otherwise the double-clockwise may be recognized twice since its duration is greater than 1.0s.

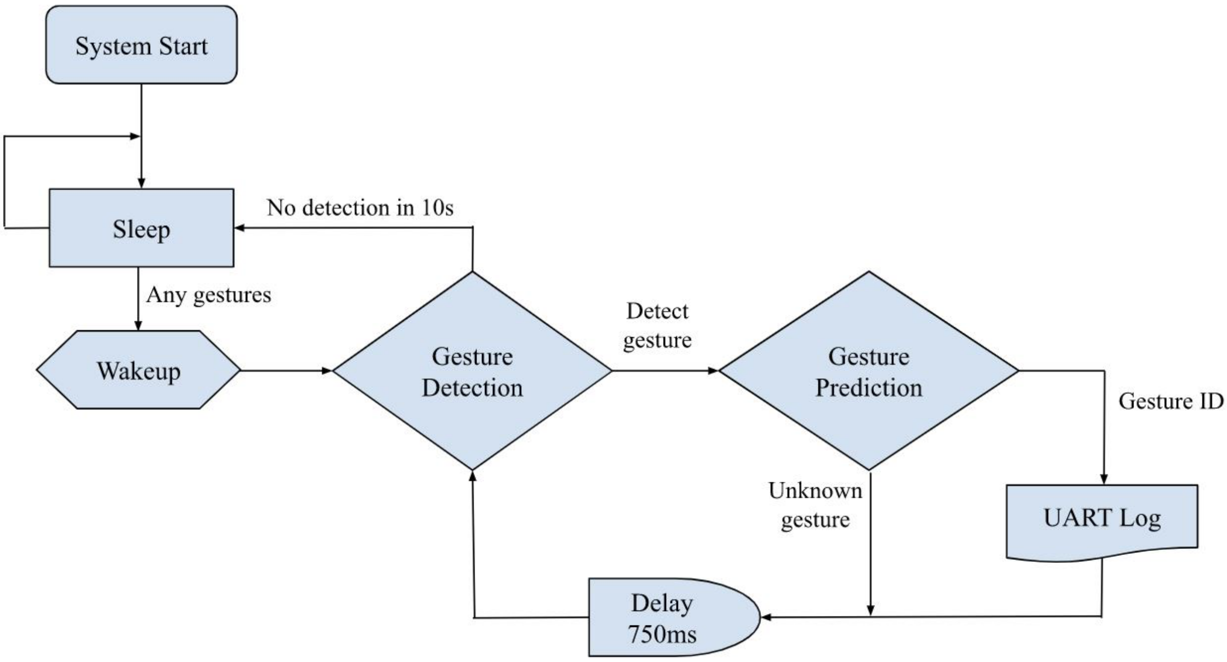


Figure 14: Firmware flow. From the system point of view, the interval between two consecutive gestures is at least 1s, we simply use a delay strategy for gesture spotting.

4.6 System performance metrics and analysis

The actual performance of the trained model is measured by counting the correct manipulation of VLC multimedia player. By arranging seven testers, five of whom are first-time users, explain and demonstrate the operation to the testers for ten minutes before the test, and let the testers practice for five minutes after the explanation, and finally let the users operate each gesture 10 times and count the number of correct manipulations. The gesture control status of Group B is shown in Figure 15. The average accuracy rate is 94.5%. Since the two first-time users are left-handed persons, the accuracy rate is about 2% lower than that of other users.

In order to further measure the robustness of millimeter-wave radar to external environmental interference as shown in Figure 16, we test it under four different types of disturbance, Bluetooth (headset), electromagnetic waves (hair dryer) and fan swiping. As shown in Figure 17, the recognition accuracy drops by less than 5%. If there is electromagnetic wave or millimeter wave interference around, the recognition accuracy will be reduced within 5%. Left-swipe and right-swipe are a bit sensitive to environmental disturbances.

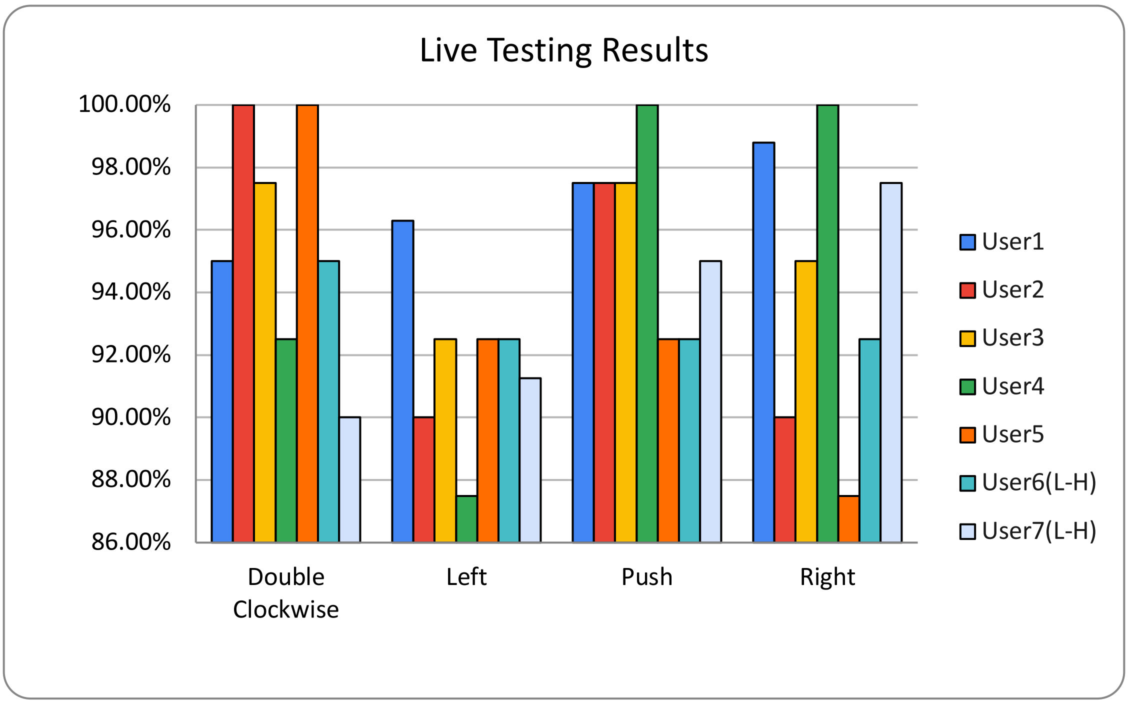


Figure 15: Group B live testing recognition accuracy results among seven users for media player application. The average accuracy rate is 94.5%.

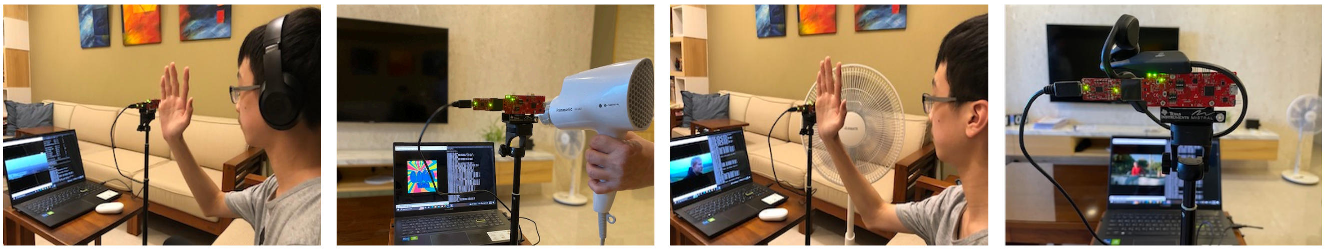


Figure 16: Four types environmental interference scenarios. From left to right are Bluetooth (headset), electromagnetic waves (hair dryer), fan swiping and millimeter-wave radar.

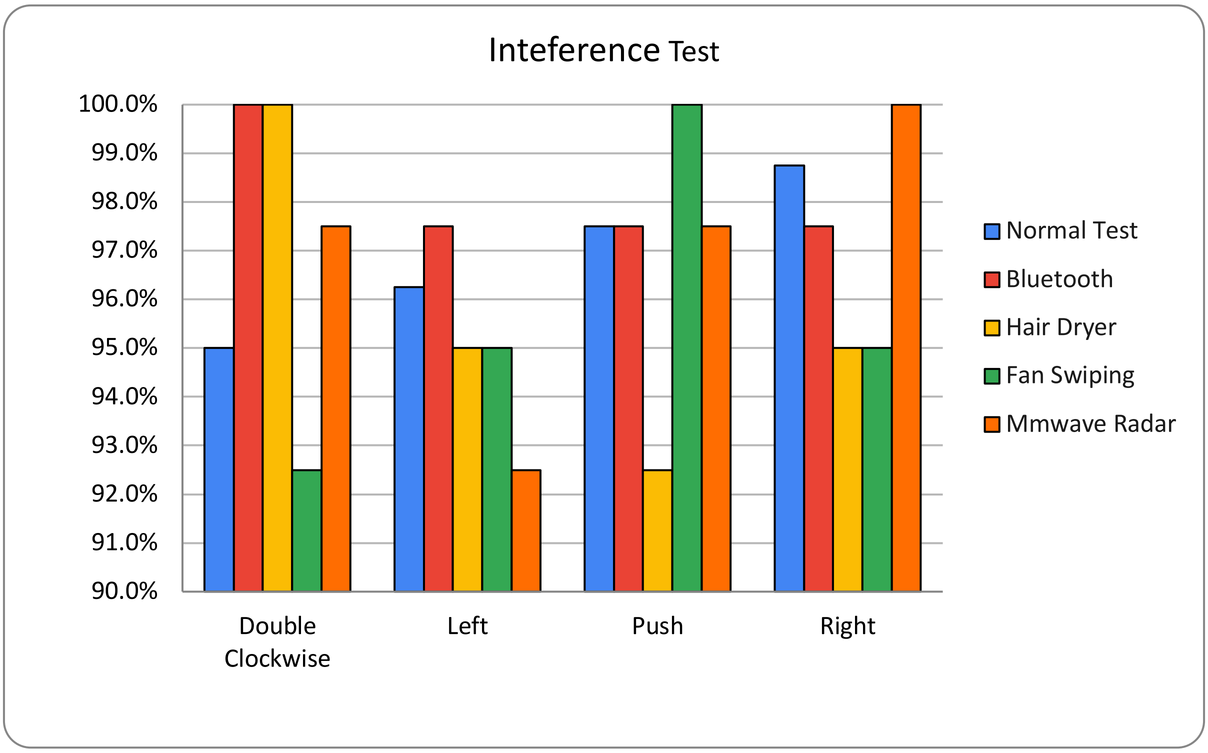
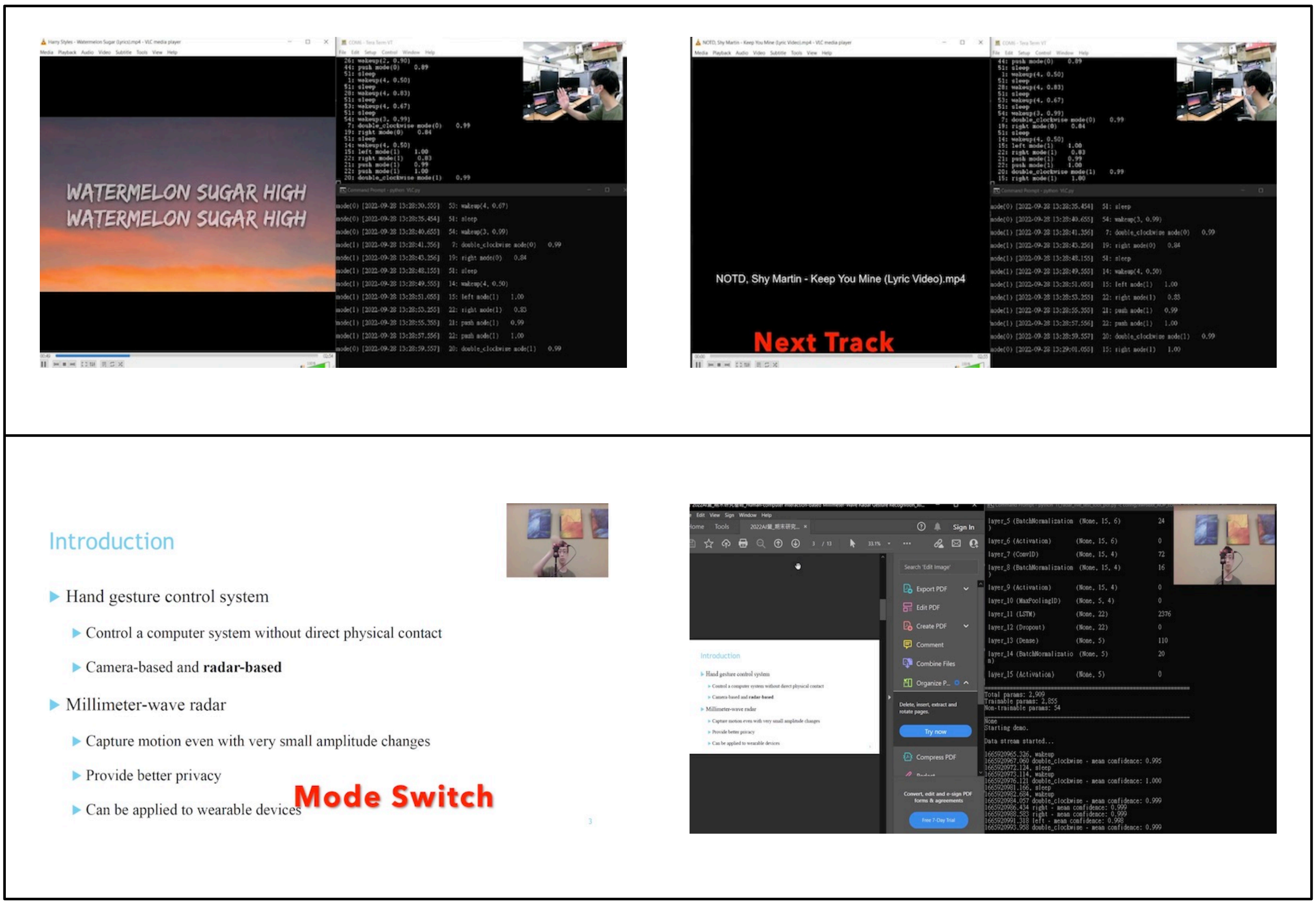


Figure 17: Recognition accuracy of different environmental interference scenarios. Accuracy drops by up to 5% in complex spaces.

5. Demo

Figure 18 is the snapshots of the gesture interaction with VLC media player and the gesture manipulation of the PDF presentation. In PDF presentation operation, we assigned left-swipe for previous slide control, right-swipe for next slide control and double-clockwise for presentation mode switching. According to the live testing results, compared with other gesture recognition accuracy in Group B, left-swipe is a bit sensitive to gesture actions, such as the hand position is lower than the radar, the gesture movement is too fast, etc.



```
26: wakeup(2, 0.90)
44: push mode(0)    0.89
51: sleep
 1: wakeup(4, 0.50)
51: sleep
28: wakeup(4, 0.83)
51: sleep
53: wakeup(4, 0.67)
51: sleep
54: wakeup(3, 0.99)
 7: double_clockwise mode(0)    0.99
19: right mode(0)    0.84
51: sleep
14: wakeup(4, 0.50)
15: left mode(1)    1.00
22: right mode(1)    0.83
21: push mode(1)    0.99
22: push mode(1)    1.00
20: double_clockwise mode(1)    0.99
```

```
mode(0) [2022-09-28 13:28:30.555]   53: wakeup(4, 0.67)
mode(0) [2022-09-28 13:28:35.454]   51: sleep
mode(0) [2022-09-28 13:28:40.655]   54: wakeup(3, 0.99)
mode(1) [2022-09-28 13:28:41.356]    7: double_clockwise mode(0)    0.99
mode(1) [2022-09-28 13:28:43.256]   19: right mode(0)    0.84
mode(1) [2022-09-28 13:28:48.155]   51: sleep
mode(1) [2022-09-28 13:28:49.555]   14: wakeup(4, 0.50)
mode(1) [2022-09-28 13:28:51.055]   15: left mode(1)    1.00
mode(1) [2022-09-28 13:28:53.255]   22: right mode(1)    0.83
mode(1) [2022-09-28 13:28:55.355]   21: push mode(1)    0.99
mode(1) [2022-09-28 13:28:57.556]   22: push mode(1)    1.00
mode(0) [2022-09-28 13:28:59.557]   20: double_clockwise mode(1)    0.99
```

```
44: push mode(0)    0.89
51: sleep
 1: wakeup(4, 0.50)
51: sleep
28: wakeup(4, 0.83)
51: sleep
53: wakeup(4, 0.67)
51: sleep
54: wakeup(3, 0.99)
 7: double_clockwise mode(0)    0.99
19: right mode(0)    0.84
51: sleep
14: wakeup(4, 0.50)
15: left mode(1)    1.00
22: right mode(1)    0.83
21: push mode(1)    0.99
22: push mode(1)    1.00
20: double_clockwise mode(1)    0.99
15: right mode(1)    1.00
```

```
mode(0) [2022-09-28 13:28:35.454]   51: sleep
mode(0) [2022-09-28 13:28:40.655]   54: wakeup(3, 0.99)
mode(1) [2022-09-28 13:28:41.356]    7: double_clockwise mode(0)    0.99
mode(1) [2022-09-28 13:28:43.256]   19: right mode(0)    0.84
mode(1) [2022-09-28 13:28:48.155]   51: sleep
mode(1) [2022-09-28 13:28:49.555]   14: wakeup(4, 0.50)
mode(1) [2022-09-28 13:28:51.055]   15: left mode(1)    1.00
mode(1) [2022-09-28 13:28:53.255]   22: right mode(1)    0.83
mode(1) [2022-09-28 13:28:55.355]   21: push mode(1)    0.99
mode(1) [2022-09-28 13:28:57.556]   22: push mode(1)    1.00
mode(0) [2022-09-28 13:28:59.557]   20: double_clockwise mode(1)    0.99
mode(0) [2022-09-28 13:29:01.055]   15: right mode(1)    1.00
```

Introduction

- Hand gesture control system
  - Control a computer system without direct physical contact
  - Camera-based and **radar-based**
- Millimeter-wave radar
  - Capture motion even with very small amplitude changes
  - Provide better privacy
  - Can be applied to wearable devices

```
layer_5 (BatchNormalization  (None, 15, 6)          24
)
layer_6 (Activation)         (None, 15, 6)          0
layer_7 (Conv1D)             (None, 15, 4)          72
layer_8 (BatchNormalization  (None, 15, 4)          16
)
layer_9 (Activation)         (None, 15, 4)          0
layer_10 (MaxPooling1D)      (None, 5, 4)           0
layer_11 (LSTM)              (None, 22)             2376
layer_12 (Dropout)           (None, 22)             0
layer_13 (Dense)             (None, 5)              110
layer_14 (BatchNormalizatio  (None, 5)              20
n)
layer_15 (Activation)        (None, 5)              0
=================================================================
Total params: 2,909
Trainable params: 2,855
Non-trainable params: 54
_________________________________________________________________
None
Starting demo.

Data stream started...

1665920965.326, wakeup
1665920967.060 double_clockwise - mean confidence: 0.995
1665920972.124, sleep
1665920973.114, wakeup
1665920976.121 double_clockwise - mean confidence: 1.000
1665920981.166, sleep
1665920982.684, wakeup
1665920984.057 double_clockwise - mean confidence: 0.999
1665920986.434 right - mean confidence: 0.999
1665920988.583 right - mean confidence: 0.999
1665920991.318 left - mean confidence: 0.998
1665920993.958 double_clockwise - mean confidence: 0.999
```

Figure 18: Snapshots of interaction with VLC media player (top) and PDF presentation manipulation (bottom).

6. Conclusion

We proposed an efficient millimeter-wave radar dynamic gesture control system for the selected application based on the user experience design principle. Through iterative gesture qualitative analysis and performance quantitative evaluation, which improved gesture usability and learning model efficiency, respectively, the algorithm achieved high accuracy (94.5% on average) on 4 hand gestures across 7 users. Additionally, we provided a real-time and power-saving radar-based gesture recognition solution to interoperate applications in complex space without significant misidentification. This work can also be extended to control car multimedia systems, wearables and smart home devices by extending the gesture set to support continuous finger gesture recognition.

7. References

[1] Ya-Ting Kao and Yen-Lin Chen (2014), “Real-time Near-distance Hand Gesture Recognition for Wearable Human Computer Interaction Devices”, *National Taipei University of Technology Master thesis*

[2] Po-Yu Hsiao and Kai-Tai Song (2017), “Gesture Recognition and its Application to Human-Robot Interaction Control”, *National Chiao Tung University Master thesis*

[3] Fang-Na Lee and Dr. Peng-Hua Wang (2020), “A REAL-TIME GESTURE RECOGNITION SYSTEM BASED ON IMAGE PROCESSING”, *Department of Communication Engineering National Taipei University Master thesis*

[4] Hayashi, E., Lien, J., Gillian, N., Giusti, L., Weber, D., Yamanaka, J., Bedal, L., & Poupyrev, I. (2021). RadarNet: Efficient gesture recognition technique utilizing a miniature radar sensor. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3411764.3445367

[5] Li Yen and Po-Hsuan Tseng (2019), “Machine Learning-Based Hand Gesture Recognition Based on mmWave Radar”, *National Taipei University of Technology Master thesis*

[6] Wang, S., Song, J., Lien, J., Poupyrev, I., & Hilliges, O. (2016). Interacting with Soli. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. https://doi.org/10.1145/2984511.2984565

[7] Jaén-Vargas, M., Reyes Leiva, K. M., Fernandes, F., Barroso Gonçalves, S., Tavares Silva, M., Lopes, D. S., & Serrano Olmedo, J. J. (2022). Effects of sliding window variation in the performance of acceleration-based human activity recognition using Deep Learning Models. *PeerJ Computer Science*, *8*. https://doi.org/10.7717/peerj-cs.1052

[8] Ahmed, S., Kallu, K. D., Ahmed, S., & Cho, S. H. (2021). Hand gestures recognition using radar sensors for human-computer-interaction: A Review. *Remote Sensing*, *13*(3), 527. https://doi.org/10.3390/rs13030527

【評語】190006

本研究是一個很好的嘗試。建議階段性目標是辨識聽障者溝通的手語，終極的目標能夠辨識遠端的手語、身體姿態語言，及辨識遠端的唇語。

此作品用 Millimeter-wave Radar 來量測四種手勢的運動資料，之後利用三種機器學習模型（CNN、 CNN-LSTM、和 CNN-BiLSTM）來進行訓練和辨識正確率的測試。技術上只是使用 Millimeter-wave Radar 來量測手部運動資料和使用一些機器學習模型來進行訓練，這些機器學習模型的技術並無改進之處。目前辨識正確率可達到 90%左右，但目前需要辨識的不同手勢只有四種因此容易達成高辨識率。建議未來增加不同手勢的數目來探討不同手勢數目對辨識率的影響。英語報告流暢自然。