J-050

Recognizing On-Surface Gesture Using Smartwatch

Junyu Chen, Hiroshi Saito, and Hiroshi Nakamura
Department of Information Physics and Computing, Graduate School of Information Science and Technology
The University of Tokyo

Abstract

With the increasing need to interact with nearby devices, smartwatches becomes an ideal device for sending commands. Meanwhile, utilizing surfaces like tables and floors around us provides us with both tactile feedback and support. Therefore, in this paper, we propose a system to recognize on-surface gestures using the accelerometer and gyroscope of a smartwatch. A 1D Convolutional Neural Network is applied for classification. We furthermore conducted a user study with a group of ten to evaluate the validity of the system, as a result, average Macro F1 scores of 0.992, 0.878, and 0.957 are reached in in-person mode, cross-person mode, and hybrid-10 mode, respectively.

1. Introduction

With the increasing pervasiveness of the electronic technologies, an abundance of smart devices like television, air conditioner, desktop have become a common sight in our home, office and public area, the convenience of the way how we interact with device become more and more important. While universal remote controller were designed to make interactions more convenient, most of them are using smartphone as a hub.

Meanwhile, as smartwatches become more popular due to their affordability and high functionality, their wearable property also makes them a better solution for interacting with nearby devices, as they are always on our wrist. This means, unlike smartphones, we don't need to take them out of our pocket every time we want to use them. However, the limitations of a smartwatch's screen size can make it inconvenient to interact with. Aside from using voice commands or camera-based methods, which cause privacy problem, much research has been done on detecting hand gestures directly using the sensors on the smartwatch. Nevertheless, many of these studies focus on mid-air interaction, which leads to another problem: mid-air fatigue and a lack of tactile feedback. One solution would be utilizing the surface around us, such as tables, walls, floors, and even our skins like arm and lap [1]. These surface are not only always around us but also provide us enough support and feedback to reduce our fatigue and make the interaction more intuitive.

Therefore, in this paper, we present a system to recognize on-surface gestures using off-the-shelf smartwatch. This is achieved using the Inertial Measurement Unit (IMU) in the smartwatch to track both hand movements and vibrations caused by collisions between fingers and the desktop. The 6-axis data are sent to a 1D Convolutional Neural Network to differentiate signal patterns and recognize gesture categories. We furthermore conducted a user study within a group of ten, finally we achieved average accuracy of 99.155% when training with participants' own data and 95.704% when training with other participants and few samples of the targeting people's data.

2. Related Work

2.1. Hand gesture detection using Smartwatch

Previous studies have shown the potential to detect hand gesture and finger gestures using the sensors on smartwatch or a wristband. For example, AirContour [2] utilizes IMU in the smartwatch for in-air hand writing by tracking arm motions. In another example, SignSpeaker [3] have developed a real time sing language translator using an off-the-shelf smartwatch. By boosting IMU sampling frequency in the smartwatch, Viband [4] can detect fine-grained hand gestures and sense grasped objects.

Besides motion signals, various other sensors have been utilized in existing works. For instance, FinDroidHR [5] can detect 10 hand gestures by measuring wrist blood volume changes through Photoplethysmography (PPG) in the smartwatch. GestEar [6] combines microphone and motions sensors to recognize sound-emitting gestures.

However, all these methods focus on mid-air gestures, which lacks tactile feedback and can lead to arm fatigue after prolonged operations [7].

2.2. On Surface interaction

Plenty of works have been done focusing on input on surface, which can be categorized as instrumented surface based methods or uninstrumented surface based methods depending on whether the surface is modified or not.

Instrumented surface based methods always instrumented surface with capacitive and electrical impedance. For example, Wall++ [8] turns ordinary walls to smart infrastructure to track user touches and poses by patterning electrodes onto a wall using conductive paint. Electrick [9] enables touch input on surface by detecting impedance change on conductive coating caused by touch. However, such methods are not always available since we need to instrument the surface before interactions.

To detect touch input without modifying surface, wear-

able motion sensors are widely used. QwertyRing [10] detects text entry on on desk-like surface by placing an IMU ring on the user's finger. By using the fact that different finger taps cause vibration propagating in different path, TapID [11] can not only detect tap events but also identify the finger used for touch using a IMU wristband. Acustico [12] enables tap detection and 2D tap localization on surfaces using IMU and microphone from a wrist-worn device. Nevertheless, these works mainly focus on tap detection and don't extend to other gestures. Unlike existing works, we concentrate on detecting different on-surface hand gestures using an off-the-shelf smartwatch, taking the advantage of support and tactile feedback from the surface without modifying it.

3. System Overview

3.1. Sensing Principle

Wearing smart watch on our wrist, not only can the hand movement cause the smartwatch to move, but the vibrations caused by our finger touched the surface can also be transmitted to the smartwatch via bones, which are then measured by the Gyro sensors in the smart watch. Therefore, even if on-surface gestures primarily involve finger movements with little hand movement, they can still be sensed by the smartwatch. And as figure 1 shows, different gestures involve distinct actions and frequencies, leading to unique patterns in the signals captured from smartwatch, which is our essential sensing principle.

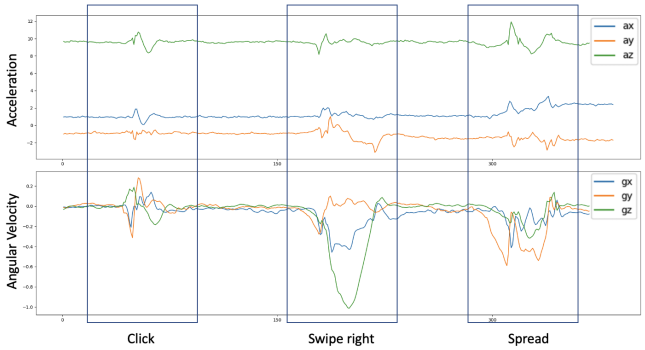


Figure 1. Signal Patterns when performing different gestures on table, upper part represents 3-axis of accelerometer and lower part indicates 3-axis of gyroscope

3.2. Model

Inspired by the network architecture in previous work [11], we also adopted a VGG-like multi-layer 1D-convolutional neural network for the classification task. Shown in figure 2, the shape of input is (128 x 6), which means 128 data points was collected and each data points

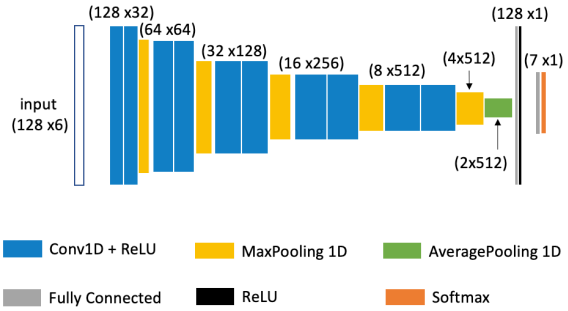


Figure 2. Neural Network Architecture for on-surface gesture classification

contains 6 channels, corresponding to 6-axis of IMU. Subsequently 5 convolutional blocks are used to extract features, each of them consists of two consecutive 1D convolutional layers with a kernel size of 3 and 1D Max pooling layer. The idea of consecutively using two convolutional layers with a small kernel size is stated in [13], which is more effective by increasing the non-linearity without affecting the receptive field. Finally features are sent one average pooling layer and two fully connected layers followed by a softmax activation functions to get the final probability distribution over 7 gestures.

4. Evaluation

To evaluate the performance of the proposed method, we conducted a user study with a group of ten participants.

4.1. Data Collection

4.1.1 Gesture Set

Without loss of generality, as figure shows, we designed 7 types of gestures commonly used in on-surface interactions, including swipe in four directions, click, pinch and spread, which we are already very familiar on trackpads.

Figure 3. Seven gestures defined in the user study, from left to right, in order there are swipe left/right, swipe up/down, pinch/spread and click.

	swipe left	swipe right	swipe up	swipe down	click	spread	pinch	Macro F1
In person	0.99	0.993	0.988	0.997	0.993	0.998	0.983	0.992
Cross person	0.929	0.961	0.908	0.896	0.788	0.892	0.773	0.878
Hybrid 5	0.967	0.983	0.943	0.964	0.907	0.897	0.878	0.934
Hybrid 10	0.96	0.985	0.975	0.972	0.938	0.949	0.921	0.957
Hybrid 60	0.988	0.993	0.985	0.992	0.99	0.993	0.99	0.99

Table 1. Average F1 scores of each gesture across participants

4.1.2 Device

We used a Samsung Galaxy Watch 4 for the user study, which is based on Google Wear OS and equipped with a 9 axis IMU, including 3 axis accelerometer, 3 axis gyroscope and 3 axis magnetometer. However, in this work, we only leveraged data from accelerometer and gyroscope. A smart watch application was developed to collect data in real time using the API provided by Google. Data was collected at a sampling rate of 100 Hz and stored locally on the smart watch. After the data collection session, data was transmitted to the computer for analysis.

4.1.3 Procedure

Ten participants were involved in the user study, including 7 males and 3 females with an average age of 24 years old, all of whom are right-hand dominant. During each collecting session, participants sat comfortably in front of a table, and wore the smart watch on their right wrist. For each gesture, examples were given by the instructor and participants first practiced a few times to familiarize themselves with the gesture. Then they were asked to perform each gesture 100 times, and they are free to stop and rest anytime during the session.

Finally, a total of 10 * 7 * 100 samples were collected, each sample last for 1.5 seconds. The shape of each sample is 150 * 6, where 150 means 150 data points since the sampling frequency is 100 Hz and 6 means each data point contains 6-axis (3-axis of gyroscope and 3-axis of accelerometer).

4.2. Pre-processing and Segmentation

Though our collected samples contain 150 data points, we found 128 data points is long enough for the classification. Therefore, to augment the dataset and make our model more robust, we applied a sliding window with a length of 128 data points and step size of 1 to segment the data, allowing us to create more fine-grained variations in the data.

Before sending data to the network, we also normalized the input with z score normalization, where standard deviation and mean is calculated from the whole dataset.

$$x_{\text{normalized}} = \frac{x - \text{mean}(x)}{\text{std}(x)}$$

4.3. Configuration

For each participants, we first divided dataset randomly into 80% as training dataset and 20% as testing dataset. The training and testing are done on Apple MacBook Pro (2023 model) with Apple Silicon M2 Pro Chip, average inference time for each input is about 4 ms.

During the training process we adopted SGD optimizer and cross entropy for the loss function.

4.4. Result

We evaluate the performance using average accuracy and average $Macro\ F_1$ score for each gesture across all participants as formula shows.

$$\text{Average Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m} accuracy(i,j)$$

$$\text{Average Macro } F_1 = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m} F_1(i,j)$$

In which n equals to the number of participants and m equals to the number of gestures. $accuracy(i,j)$ and $F_1(i,j)$ means the classification accuracy and F_1 score for gesture j of participant i. $Macro\ F_1$ score means for each participant, we treat each gesture equally by calculating their F_1 score independently and then take the unweighted mean across all gestures.

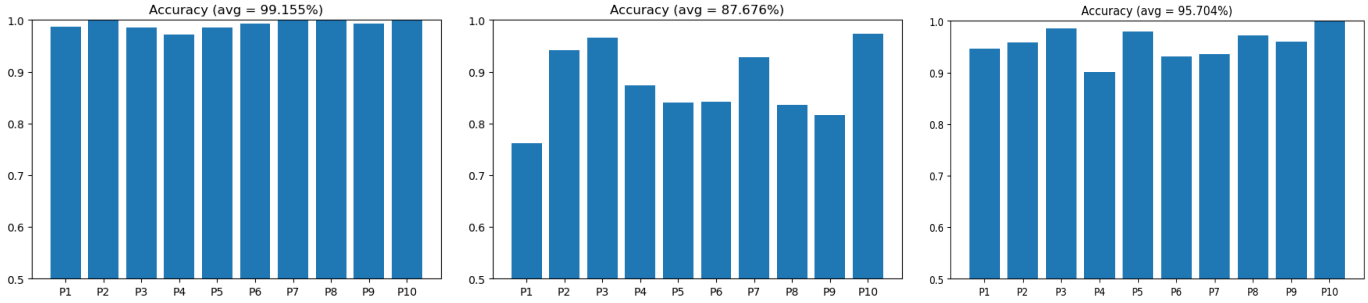
To get a more comprehensive result, we evaluate the result in three different modes: in-person, cross-person and hybrid, using different fraction of data.

In Person

We built a single model for each participants using his/her own data. Their own training dataset was used for training and testing dataset for testing. As figure 4a and table 1 shows, an average accuracy of 99.155% and average $Macro\ F_1 = 0.992$ is reached.

Cross Person

In this mode, We performed a 10 fold cross-validation, testing on each participant's testing dataset and training on all



(a) In-Person Mode (b) Cross-Person Mode (c) Hybrid-10 Mode

Figure 4. Average classification accuracy of each participant in different mode

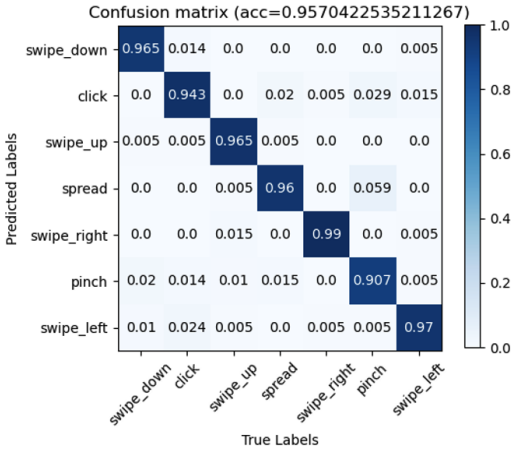


Figure 5. Average Confusion matrix in hybrid-10 mode

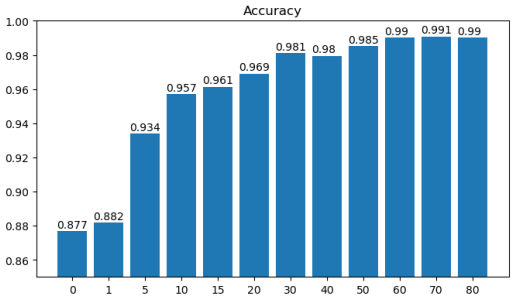


Figure 6. Average classification accuracy in hybrid-n mode

other people's training dataset. The average accuracy is 87.676% and average $Macro\ F_1 = 0.878$.

Hybrid Mode

In hybrid mode, we also performed the 10 fold cross-validation combining part of the target peoples' data, i.e., for each participants we tested on their own testing dataset, trained on few samples from their training dataset and all other people's training dataset. This mode indicates the performance when a new user comes to the system and fine-tuning using little of his/her data. As figure 4c shows, when we add 10 samples of each gestures from targeting people to training dataset, The average accuracy is 95.704% and average $Macro\ F_1 = 0.957$, figure 5 depicts the confusion matrix in the condition.

We further evaluated hybrid mode with different amount of targeting people's data involved, the result is shown in figure 6, where number n means n samples of each gestures from targeting people are used for training. For example, hybrid-1 means 1 sample of each gesture from targeting user's training dataset and all other people's training dataset are used for training, and an accuracy of 88.2% is achieved.

4.5. Discussion

The result confirmed the validity of using our system to recognize on surface gestures. When the model is built using user's own data, average accuracy is up to 97.959%, which is high enough for practical use.

Unsurprisingly, when the model is totally built on other people's data, the performance is much worse than in person mode. However, with little data of the targeting people data added, the performance is again high enough, showing the generalization ability of our system with little fine-tuning. In detail, accuracy increased rapidly as first 5 or 10 samples added, then increased gradually and reached its highest point when 60 samples of each gestures were added.

Taking a look at the performance of individual gestures, we further noticed that swipe left and right got a much higher F_1 score even in cross person mode, we think this is because the amplitude of our hand movement is much larger when performing swipe left and right, making it easier to recognize than other subtle finger movement like click.

5. Future Work

Though our initial study showed a promising result that it is feasible to detect different types of on surface gestures us-

ing smartwatch, there are still challenges remaining to make the system work in real life.

Including more participants and gestures

Presently, our method can only distinguish seven kinds of different gestures within a group of ten. Although we got a high average score on both in person mode and hybrid mode, for some participant, classification accuracy is clearly lower than others, which shows even for the same gesture, different people will have different patterns. In the future, we plan to address this problem by collecting more data within a larger user group performing more kinds of gestures as our base model, such that it will contain more variants and increase the generalization ability.

Extending to New Surface

The current model is trained using data all collected in the same condition, which is right hand based and the surface is table. However, the real life scenario may change case by case, for example, some people may prefer to use left hand, or sometime tables is not available and they have to interact with other surface like wall and floors. These scenario is very common and our current method won't work anymore since signal pattern is totally different.

One possible solution is applying techniques like few shots learning in the ongoing research, which is training a model to tell the similarity of two input signals instead of directly mapping signals to categories. As a result, the user only needs to collect a few samples of the new condition as a reference. Each time a new real-time signal comes to the system, it will be compared with all references and will be categorized as the one with the most similarity. No extra training is required, and the user's burden is much relieved as they only need to collect a few samples.

6. Conclusion

We have presented a system to detect on surface hand gestures using IMU sensor from off-the-shelf smartwatch. Through a 10-participant study, we further demonstrate the validity of the system, with average Macro F1 scores of 0.992, 0.878, and 0.957 are reached in in-person mode, cross-person mode, and hybrid-10 mode, respectively. We believe our method can provide a new interaction way on uninstrumented surfaces.

References

[1] Chris Harrison. Appropriated interaction surfaces. *Computer*, 43(06):86–89, 2010.

[2] Yafeng Yin, Lei Xie, Tao Gu, Yijia Lu, and Sanglu Lu. Aircontour: Building contour-based model for in-air writing gesture recognition. *ACM Trans. Sen. Netw.*, 15(4), oct 2019.

[3] Jiahui Hou, Xiang-Yang Li, Peide Zhu, Zefan Wang, Yu Wang, Jianwei Qian, and Panlong Yang. Signspeaker: A real-time, high-precision smartwatch-based sign language translator. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[4] Gierad Laput, Robert Xiao, and Chris Harrison. Viband: High-fidelity bio-acoustic sensing using commodity smartwatch accelerometers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 321–333, New York, NY, USA, 2016. Association for Computing Machinery.

[5] Yu Zhang, Tao Gu, Chu Luo, Vassilis Kostakos, and Aruna Seneviratne. Findroidhr: Smartwatch gesture input with optical heartrate monitor. 2(1), mar 2018.

[6] Vincent Becker, Linus Fessler, and Gábor Sörös. Gestear: Combining audio and motion sensing for gesture recognition on smartwatches. In *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, ISWC '19, page 10–19, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. Modeling cumulative arm fatigue in mid-air interaction based on perceived exertion and kinetics of arm motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3328–3339, New York, NY, USA, 2017. Association for Computing Machinery.

[8] Yang Zhang, Chouchang (Jack) Yang, Scott E. Hudson, Chris Harrison, and Alanson Sample. Wall++: Room-scale interactive and context-aware sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–15, New York, NY, USA, 2018. Association for Computing Machinery.

[9] Yang Zhang, Gierad Laput, and Chris Harrison. Electrick: Low-cost touch sensing using electric field tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 1–14, New York, NY, USA, 2017. Association for Computing Machinery.

[10] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. Qwertyring: Text entry on physical surfaces using a ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), dec 2020.

[11] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. Tapid: Rapid touch interaction in virtual reality using wearable sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 519–528, 2021.

[12] Jun Gong, Aakar Gupta, and Hrvoje Benko. Acustico: Surface tap detection and localization using wrist-based acoustic tdoa sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 406–419, New York, NY, USA, 2020. Association for Computing Machinery.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.