Thumb Gesture Recognition Method Using Wrist EMG Signals with a Machine Learning Algorithm

Ting-Wei Chen (1), Pai-Chi Li (1, 2), Yu-Sheng Zeng (3), Yuan-Bin Lu (3), Chin-Lin Lee (3), Jiun-Jie Tsai (3), Chun-Yi Chou (3), Ching-Chun Lin (3)

1) Graduate Institute of Biomedical Electronics and Bioinformatics, National Taiwan University, Taipei, Taiwan

2) Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
3) Novatek Microelectronics Corporation, Hsinchu, Taiwan

Abstract

This paper explores using multiple electrodes on the back of a smartwatch at the wrist to capture EMG signals and employs machine learning methods to detect various thumb movements. The study results show that even with a limited number of electrodes placed only at the wrist on the back of the watch, appropriate machine learning techniques can still achieve a recognition accuracy of around 99% for thumb movements.

Author Keywords

Thumb gesture classification, Wrist EMG signals, Individual EMG data, Machine learning.

1. Introduction

Novatek has been cultivating the TDDI touch domain for many years, maintaining a leading position in the mobile phone sector. In recent years, the application range of TDDI has been extended to automotive, notebooks, tablets, and wearables, combining display output and touch input functions to provide customers with comprehensive solutions for human-machine interfaces. With the introduction of the metaverse concept, traditional human-machine interface concepts may face changes. While display output remains necessary, the input part requires the adoption of methods different from traditional touch methods, such as motion sensing input [1-2] and gesture detection [3-4].

The current control methods for the metaverse mainly involve motion-sensing touch, with technologies broadly categorized into three types: IMU (Inertial Measurement Unit) [5], optical image analysis [6-7], and muscle motion detection. The more commonly seen technology is using IMU for gesture detection. However, there are currently no commercially available products or IC solutions that use EMG (Electromyography) combined with wearables for gesture detection. Only armband-type products are available, and the number of measurement electrodes is higher than four, with no products achieving gesture detection using a small number of electrodes.

In the field of human-machine interface input, since Novatek already has a wearable TDDI touch solution, adding EMG detection [8] capability could further enhance gesture detection capabilities, making the overall input solution more complete. The goal of this research is to achieve wrist gesture detection (thumb moving left/center/right and clicking) by measuring EMG signals with a small number of electrodes (<=3) on the back of a watch (excluding the strap).

2. Methods

This study presents a machine learning model for classifying thumb gestures. The training workflow is illustrated in Figure 1.

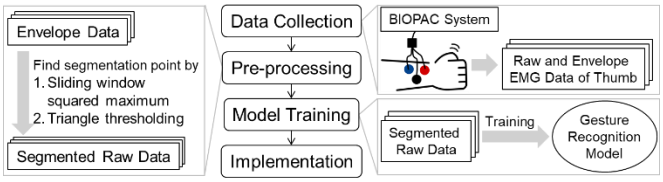

Figure 1. Workflow of the thumb gesture recognition model training process.

1. ***Data Collection***: A set of 12 thumb gestures was collected and organized into 4 classes:
   a. *Origin*: The right hand rests flat on a surface, with the fingers forming a "C" shape and the thumb tip pressing on the proximal joint of the index finger.
   b. *Leftward (Label 1; Class 1)*: The right thumb moves from the origin to the index finger's tip at a standard speed.
   c. *Rightward (Label 2; Class 2)*: The right thumb moves from the origin to the index finger's metacarpal joint at a standard speed.
   d. *Clicking (Label 3; Class 3)*: The right thumb moves vertically upward from the origin and returns to the origin at a standard speed.
   e. *Joystick Gestures (Label 4–6; Class 1–3)*: Using an auxiliary tool, the leftward, rightward, and clicking gestures are each performed and recorded.
   f. *Ineffective Gestures (Label 7–12; Class 4)*
      i. Left-Upward: The right thumb moves diagonally upward from the origin to above index finger's tip.
      ii. Right-Upward: The right thumb moves diagonally upward from the origin to above index finger's

ISSN 0097-996X/25/5601-0278-$1.00 © 2025 1

metacarpal joint.

iii. Static: The right thumb remains stationary at the origin.

iv. Slow Leftward: The right thumb moves slowly from the origin to the index finger's tip.

v. Slow Rightward: The right thumb moves slowly from the origin to the index finger's metacarpal joint.

vi. Slow Clicking: The right thumb moves slowly upward and downward from the origin.

Each gesture was completed in approximately 3 seconds. The experimental setup utilized the BIOPAC Student Lab MP36 system in EMG mode, capturing raw and envelope (ENV) signals. Data were sampled at 500 Hz with a 24-bit A/D resolution, covering an EMG frequency range of 10–250 Hz. Electrode placement followed the thumb muscle distribution, with two electrodes on the target muscle bundle at the right wrist and one ground electrode on an adjacent non-target muscle, as depicted in Figure 2.

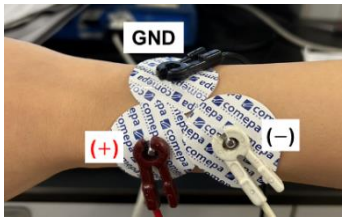

Figure 2. Electrode placement on the right wrist.

2. ***Data Pre-processing***: To optimize GPU memory usage during model training, two segmentation methods were tested and compared to reduce each signal from 1500 to 800 data points:

a. Sliding Window Squared Maximum: This method calculates the squared energy sum within a sliding window over the ENV signal. The segment is identified as the window with the highest energy.

b. Triangle Thresholding: This approach determines a threshold by connecting the peak of the ENV signal to its maximum energy point, then locating intersections from the beginning and end of the signal. These intersection points are extended evenly to complete the segment length.

The segmentation points identified by both methods were then applied to the raw data. Figure 3 demonstrates the segmentation results for both methods, illustrating differences in segment placement. Additionally, data augmentation was performed with shift intervals set at 10, 20, and 50 points to evaluate the impact of varying intervals on model accuracy during machine learning.

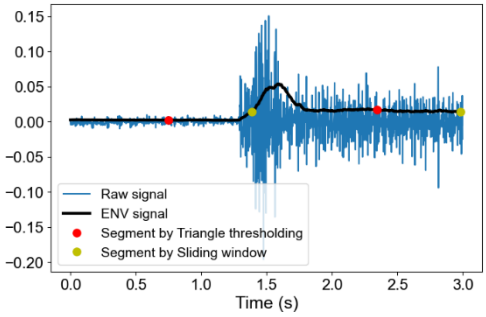

Figure 3. Example of segmentation results for both methods on the ENV signal.

3. ***Machine Learning***

a. *Dataset*: The group dataset consisted of data from 14 participants, with 10 completing all 12 gesture labels and the remaining 4 completing only labels 1–6. Prior to data augmentation and excluding joystick gestures, the sample counts per class were 140, 140, 140, and 600, respectively. For the individual dataset, a single participant performed six sessions over different days, yielding class sample counts of 121, 111, 133, and 670 for classes 1–4, excluding joystick gestures. Detailed sample counts are presented in Table I.

Dataset	Sample counts per class [class 1, class 2, class 3, class4]
1	[59, 60, 60, 319]
2	[15, 15, 19, 81]
3	[15, 15, 16, 95]
4	[10, 10, 12, 51]
5	[11, 10, 15, 60]
6	[11, 10, 11, 64]

Table I. Sample counts for each individual dataset.

b. *Model architecture*: The machine learning model was developed using K-fold cross-validation and hyperparameter tuning to optimize performance. As illustrated in Figure 4, a one-dimensional convolutional neural network (1D-CNN) architecture was selected. Various hyperparameters were tuned, including data augmentation shifts (10, 20, and 50 points), the number of CNN layers (from 1 to 5), the number of fully connected layers (from 1 to 3), and batch sizes (32, 64, and 150). Model training was performed with the Adam optimizer at a learning rate of 0.001 over 10 epochs, with an 85:15 training-to-validation split. A 10-fold cross-validation was conducted to determine the optimal hyperparameter configuration, which consisted of a 20-point augmentation shift, a 4-layer CNN, 2 fully connected layers, and a batch size of 64. All algorithms were executed on a workstation equipped with an Intel Core i9-9900KF CPU and an NVIDIA GeForce RTX 2080 Ti GPU.

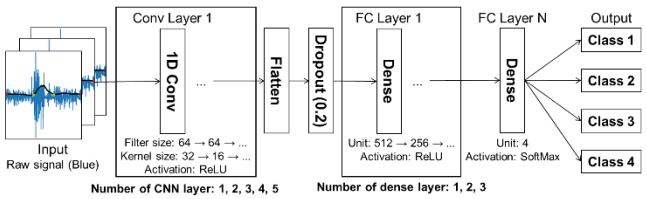

Figure 4. 1D-CNN model architecture.

3. Results

1. ***Method Comparison***: Table II shows that training the model with individual dataset significantly improved accuracy compared to the group dataset. Additionally, removing joystick gestures from the training set further improved accuracy, likely due to the distinct signal characteristics between actions with and without the auxiliary tool, which affected the model's learning efficacy. Data augmentation also enhanced model performance, whereas the choice of segmentation method had minimal impact. Based on these findings, the optimal model training approach involved using individual data without joystick gestures, combined with the triangle thresholding segmentation method and data augmentation.

Dataset	Segmentation method (S/T)	Augmentation (Y/N)	Use of auxiliary data (Y/N)	Accuracy
Individual	S	Y	N	0.982
	T	Y	N	0.981
	S	N	N	0.974
	T	N	N	0.870
	T	Y	Y	0.844
	S	Y	Y	0.786
	S	N	Y	0.723
	T	N	Y	0.638
Group	S	Y	N	0.895
	T	Y	N	0.872
	T	Y	Y	0.822
	S	Y	Y	0.779
	T	N	N	0.765
	S	N	N	0.764
	T	N	Y	0.611
	S	N	Y	0.528

Table II. Performance comparison of the model across various dataset configurations, segmentation methods, and the inclusion/exclusion of joystick gestures. S: Sliding window squared maximum; T: Triangle thresholding.

2. ***Individual Training Results***: The confusion matrices in Figure 5 show the results of a model trained on individual dataset 1 and tested on datasets 2–6. Figure 5(a) and Figure 5(c) show similar misclassification patterns, suggesting that the leftward and clicking gestures in datasets 2 and 4 share similar signal features, though distinct from those in dataset 1. Figure 5(b), Figure 5(d), and Figure 5(e) demonstrate an average accuracy of 0.934, indicating that signal features in datasets 3, 5, and 6 are more closely aligned with those in dataset 1.

Figure 6 displays confusion matrices for models trained on various combinations of datasets and tested on the remaining sets. Figure 6(a) and Figure 6(b) further confirm the similarity between datasets 2 and 4, while Figure 6(c) through Figure 6(e) show a noticeable increase in accuracy as more datasets are included in training. This improvement demonstrates that expanding the training set with multiple datasets enhances the model's generalization and classification performance.

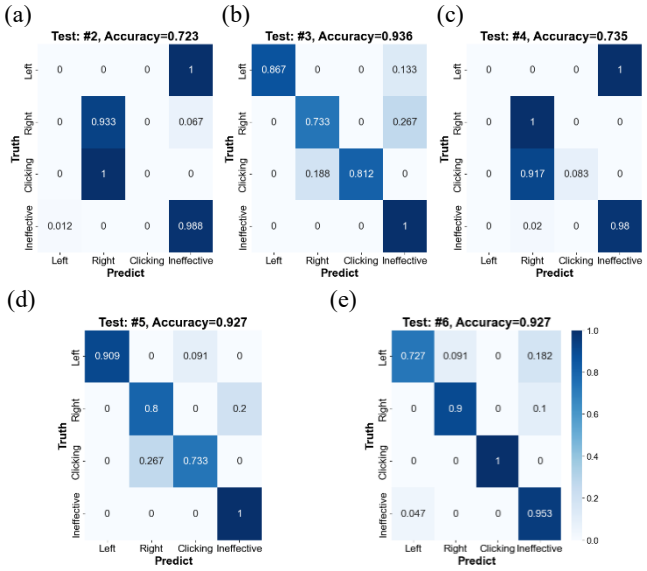

Figure 5. Confusion matrices for models trained on individual dataset 1 and tested on datasets 2–6.

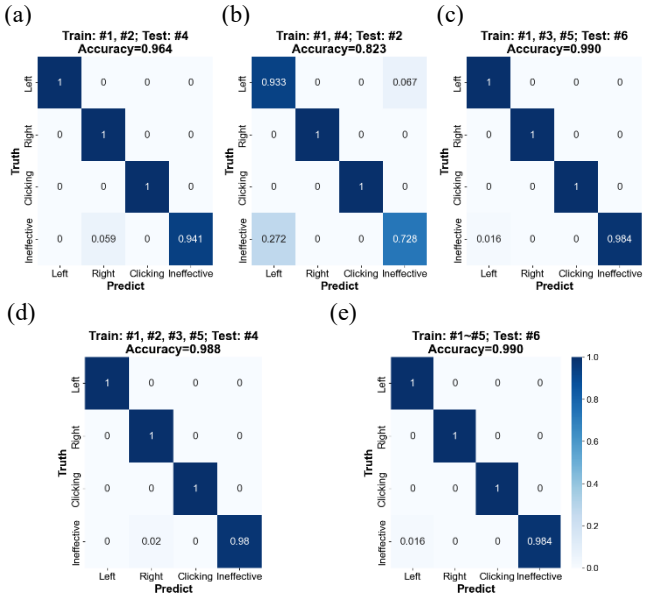

Figure 6. Confusion matrices for models trained on combinations of individual datasets and tested on remaining datasets.

4. Conclusion

In the early stages of the study, the model was trained on EMG signals from a subset of individuals and applied to infer gestures from different individual's EMG signals. Under this framework, the best accuracy achieved was 82%. Previous literature [9-10] indicates that EMG signals have strong individual variability. By adjusting the model to conduct training and inference on individual

EMG data, and incorporating augmentation, the accuracy significantly improved to 95%~99%.

Unlike common arm-based EMG gesture recognition devices, we focus on gesture recognition using wrist EMG signals. The ultimate goal is to integrate with Novatek's smartwatch wearable TDDI IC-NT38350. This study successfully demonstrates the feasibility of detecting EMG gestures at the smartwatch position.

References

1. Jang, D. K., Yang, D., Jang, D. Y., Choi, B., Jin, T., & Lee, S. H. (2023, October). MOVIN: Real-time Motion Capture using a Single LiDAR. In *Computer Graphics Forum* (Vol. 42, No. 7, p. e14961).
2. Ha, E., Byeon, G., & Yu, S. (2022). Full-body motion capture-based virtual reality multi-remote collaboration system. *Applied Sciences*, *12*(12), 5862.
3. Duan, S., Zhao, F., Yang, H., Hong, J., Shi, Q., Lei, W., & Wu, J. (2023). A pathway into metaverse: Gesture recognition enabled by wearable resistive sensors. *Advanced Sensor Research*, *2*(8), 2200054.
4. Ji, B., Wang, X., Liang, Z., Zhang, H., Xia, Q., Xie, L., ... & Yin, E. (2023). Flexible strain sensor-based data glove for gesture interaction in the metaverse: a review. *International Journal of Human–Computer Interaction*, 1-20.
5. Kim, M., Cho, J., Lee, S., & Jung, Y. (2019). IMU sensor-based hand gesture recognition for human-machine interfaces. *Sensors*, *19*(18), 3827.
6. Javier Traver, V., Latorre-Carmona, P., Salvador-Balaguer, E., Pla, F., & Javidi, B. (2014). Human gesture recognition using three-dimensional integral imaging. *Journal of the Optical Society of America A*, *31*(10), 2312-2320.
7. Czuszyński, K., Rumiński, J., & Kwaśniewska, A. (2018). Gesture recognition with the linear optical sensor and recurrent neural networks. *IEEE Sensors Journal*, *18*(13), 5429-5438.
8. Kim, J., Mastnik, S., & André, E. (2008, January). EMG-based hand gesture recognition for realtime biosignal interfacing. In *Proceedings of the 13th international conference on Intelligent user interfaces* (pp. 30-39).
9. Hsu, Y., & Young, K. (2007). An EMG-Based Dynamic Motion Prediction Mechanism Using Recurrent Neural Networks. Institutional Repository of NYCU. https://ir.lib.nycu.edu.tw/handle/11536/38244
10. Guidetti, L., Rivellini, G., & Figura, F. (1996). EMG patterns during running: Intra-and inter-individual variability. *Journal of Electromyography and Kinesiology*, *6*(1), 37-48.