Utilising Mobile Phone Sensor Gyroscopic Data For Classifying Common Hand Gestures

Ashley Hunt   
Computer Science  
University of Nottingham

Amit Kumar  
Data Science  
University of Nottingham

Shreeya Kumbhoje  
Data Science  
University of Nottingham

*Abstract*—Abstract to go here.

# Introduction

AI usage has grown over the year and will continue to grow in the near future. Mostly businesses already using AI and this has led to researcher exploring to do more research on application of AI in diverse fields.

Gesture recognition is one such application which basically is making the computer understand how physical activities are conducted among human beings. This gesture input to computers offers numerous benefits, including monitoring activities and vital signs, enhancing gaming experiences, and supporting applications in robotics and wearable technology. This technology enables devices to understand and react to human gestures, facilitating control and communication.

We are using hand gesture recognition for ( choose one: robotics/ Gaming/ Sign Language translation, etc(choose from the Applications where individuals can use smartphones to collect hand gesture data? Or whatever u feel )

# Background

"The highest performance was achieved using a convolutional neural network with long- short term memory (CNN-LSTM)"

"The pre-processing step consisted of a fourth-order low-pass Butterworth filtering with a cut-off frequency of 5 Hz. Such frequency was chosen due to the frequency content of the acquired signals, which was below 5–6 Hz, as verified by Fourier power spectrum analysis. The pre-processing also included raw signal standardization: each signal acquired with the MMR sensors was centered to have zero mean and standard deviation equal to one."

"Concerning ML approaches, inspired by human activity recognition work in the literature, the following classifiers were evaluated: Support Vector Machine (SVM) [19,20], Random Forest (RF) and K-Nearest Neighbour (KNN)."

"The complete dataset, composed of 22 subjects, was divided into training/validation and test sets using leave-one-subject-out cross validation. As concerns ML methods, the training/validation dataset was split into 5 folds, and grid search-based 5-fold cross validation was used for tuning the hyperparameters. As concerns DL methods, validation was performed on 20% of the training/validation set."

"In multi-gesture classification, CNN and CNN-LSTM performed better than ML methods, with CNN-LSTM resulting in the highest balanced accuracy"

"In the binary classification, the highest f1-score and precision values were obtained with CNN" (Moccia, Sara et al.)

"Support vector machine (SVM) and hidden Markov model (HMM) are two typical machine learning methods in pattern recognition."

"SVM usually needs careful feature extraction and selection, as well as other traditional ML algorithms like naïve Bayes (NB), K-nearest neighbors (KNN), and decision tree (DT)"

"For time sequential signals, recurrent neural networks (RNN) allow information to persist. Thus, we can make full use of the contextual information of the sequence. However, RNN might suffer from a vanishing gradient problem with long data sequences. To solve this problem, Hochreiter et al. designed a special RNN, the long short-term memory (LSTM) network that can learn long dependencies" (Zhao, Shenglin et al.)

# Metholody

Methodology Introduction goes here.

## Data Collection

Data was collected using the Phyphox mobile application[[1]](#footnote-1). Each member of the team attempted to perform the same four gestures for eight iterations per recording, and five recordings each per gesture – 40 iterations per person per gesture for a total of 480 complete gestures. These gestures were initially recorded continuously, without stopping, as per the coursework specification but after trial and error we discovered that segmenting the data was a lot more difficult when there were no clear spaces between the gestures. Instead, we opted to leave a distinct one second gap between performing each gesture in each recording. Adding these pauses allowed us to segment the recording more easily into its separate gesture patterns.

## Data Pre-Processing

Once the data was recorded and the associated csv files were imported into pandas *DataFrame* we began to preprocess the data. First, we applied a scikit-learn *MinMaxScaler* to the data – which transforms the data to values between 0 and 1 whilst maintaining the relationships between the datapoints, this is done using the below formula:

We performed this transformation to ensure that the machine learning models did not perceive higher values as more important for training.

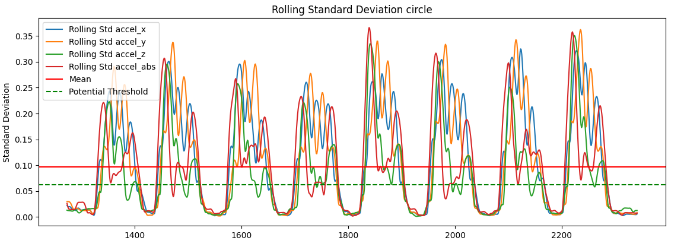
Next, we applied a basic trim to the data. Inspecting, from the beginning of the file, every 20 data points. If the mean acceleration was < 0.3 then the data was removed. We did this from the beginning of the file as well as the end of the file. This would help us reduce unnecessary noise at the beginning and end of the file, making segmentation easier.

Once the data was normalized and trimmed, utilizing the *scipy.signal* *butter* and *filtfilt* functions, we applied a low-pass frequency filter to the data. Essentially this would apply some dampening to the sensor data and reduce some of the noise in the system. Visually, this appeared to make the waveforms smoother. This was done to ensure that the model was not simply reading all the noise in the system but understanding the more generic shape of the data.

Additional signal processing methods were evaluated such as fourier transformation, dynamic time warping and wavelet transformation. Fourier transformation can be used to decompose a signal into its underlying frequencies which can be useful for removing noise from the gyroscopic data as well as determining the most important parts of the signal that define the gestures. Dynamic time warping can be used to find the similarities/distance between different waves, which can be used to shift rotate the waves so that slight misalignments in the time domain are not factors in the segmentation and eventual feature extraction of the gestures.

## Segmenting the Data

The next step of processing was to separate the individual gestures from the files that contained several gestures of the same type. This would be required so that each gesture was separated in its own block of data to be associated with a label. To apply this separation first we would visually inspect the data and its rolling average and deviation – this would help us identify a good value to decide on when one gesture was starting, and another was ending.



This figure outlines our suggested splitting point (any points where the acceleration standard deviation is lower than the threshold, in green). Using these points, we could segment the file into its separate gestures ready for feature extraction.

## Feature Extraction

For the model to accept our data we would need to transform the data once again so that each gesture’s data was the same shape. This means that we could not have varying lengths of data for a gesture. To achieve this requirement, we would segment the data once again, this time with each gesture being broken up into distinct sections with some overlap. Then using these slices of data, we could extract key features such as: mean, standard deviation', media', skew and kurtosis. This would result in data that is a summary of each slice of the gesture. These features could then be visualized and taken through the feature selection process.

## Feature selection

Once all features were extracted, we visually inspected the averages of the features to find the features that had the clearest disparity between the gestures. Feature selection is critical in order to reduce the computational and time cost of classification, as well as increase the model’s accuracy.

A graph of different colored lines

Description automatically generated

A graph with different colored lines and dots

Description automatically generated

A graph with colorful lines and dots

Description automatically generated

The features that provided clear separation between the gestures were selected and the features that did not provide strong correlation were dropped.

1. Table Type Styles

| Table Head | EXAMPLE TABLE HEADER | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

# Results

Results go here.

# Discussions

Discussion goes here.

# Conclusion

Conclusion goes here.

# References

1. Akhavian, R. and Behzadan, A.H. (2015) ‘Construction equipment activity recognition for simulation input modeling using mobile sensors and machine learning classifiers’, *Advanced engineering informatics*, 29(4), pp. 867–877. Available at: https://doi.org/10.1016/j.aei.2015.03.001.
2. Moccia, Sara et al. “Automated Classification of Hand Gestures Using a Wristband and Machine Learning for Possible Application in Pill Intake Monitoring.” Computer methods and programs in biomedicine 219 (2022): 106753–106753. Web.
3. Odhiambo, Chrisogonas Odero et al. “Detecting Medication-Taking Gestures Using Machine Learning and Accelerometer Data Collected via Smartwatch Technology: Instrument Validation Study.” JMIR human factors 10 (2023): e42714–e42714. Web.
4. Zhao, Shenglin et al. “Hand Gesture Recognition on a Resource-Limited Interactive Wristband.” Sensors (Basel, Switzerland) 21.17 (2021): 5713. Web.
5. Siddiqui, Uzma Abid et al. “Wearable-Sensors-Based Platform for Gesture Recognition of Autism Spectrum Disorder Children Using Machine Learning Algorithms.” Sensors (Basel, Switzerland) 21.10 (2021): 3319. Web.
6. Yu, Yi et al. “Exploration of Chinese Sign Language Recognition Using Wearable Sensors Based on Deep Belief Net.” IEEE journal of biomedical and health informatics 24.5 (2020): 1310–1320. Web.
7. Han H, Yoon SW. Gyroscope-Based Continuous Human Hand Gesture Recognition for Multi-Modal Wearable Input Device for Human Machine Interaction. Sensors (Basel). 2019 Jun 5;19(11):2562. doi: 10.3390/s19112562. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6603535/>
8. Stancic, I. et al. (2022) ‘Comparison and Evaluation of Machine Learning-Based Classification of Hand Gestures Captured by Inertial Sensors’, Computation, 10(9), p. 159. Available at: <https://doi.org/10.3390/computation10090159>.

1. https://phyphox.org/ [↑](#footnote-ref-1)