IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2019, accepted January 15, 2020, date of publication January 20, 2020, date of current version February 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967770

SmartGe: Identifying Pen-Holding Gesture With Smartwatch

HONGLIANG BI, JIAN ZHANG, AND YANJIAO CHEN, (Member, IEEE)
School of Computer Science, Wuhan University, Wuhan 430072, China
Corresponding author: Yanjiao Chen (chenyj.thu@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grants 61972296 and Grant 61702380, in part by the Wuhan Advanced Application Project under Grant 2019010701011419, and in part by the Hubei Provincial Technological Innovation Special Funding Major Projects under Grant 2017AAA125.

ABSTRACT Performing the correct pen-holding gesture plays an important role in handwriting efficiency and quality, especially for early education. In this paper, a detailed design and evaluation of the system, called SmartGe, is presented, which can identify the pen-holding gesture with smartwatch when writing Chinese and English. We firstly analyze the hand movement and propose a novel handwriting detection algorithm to segment each stroke or letter. Then we recognize the pen-holding gesture using deep convolution neural network(DCNN). To improve system performance in Chinese writing, we connect a vertical stroke and a horizontal stroke for pen-holding gesture recognition. SmartGe provides a convenient and natural way to improve users' writing habits, which is a lightweight system, and extensive experiments confirm its effectiveness and robustness.

INDEX TERMS Smartwatch, handwriting detection, pen-holding gesture, combined signal, deep convolution neural network.

I. INTRODUCTION

As we know, English and Chinese are two of the most commonly spoken languages in the world. How to write these two languages with the correct pen-holding gesture is an important topic. As shown in Fig. 1, there are 9 typical pen-holding gestures: correct, close grip, fold grip, tuck grip, squeeze grip, hook grip, wrap grip, mount grip and tripod grip [1]. There are fine-grained differences between correct gestures and eight wrong gestures. The thumb, the index finger and the middle finger form an open triangle, and the pen rests on the middle finger, which constructs the correct pen-holding gesture. The ideal distance between the tip of the pen and the thumb with correct gesture is around 1 inch [2], which requires less efforts in the process of writing. The correct pen-holding gesture makes users feel a stable and flexible way to write [3]. And incorrect gestures will lead to excessive forces, reducing the writing speed and even arthritic conditions and myopia [1]–[4]. Therefore, it can yield sustained benefit by taking up the right pen-holding gesture early in life.

People often find out whether the words are spelled correctly, but it's difficult to find out whether they hold the pen correctly. Existing techniques to help people correct their pen-holding gestures often need auxiliary systems. These systems force people to fix their hands in a plastic tool in a certain way [5]. It leads to uncomfortable in the process of writing and can not detect the original pen-holding gesture. Once these auxiliary devices are removed, it is easy for people to return to the wrong pen-holding gesture. There is also a recognition system that can detect pen-holding gesture by installing sensors on the pen, such as MTPen [6]. But this invisibly increases the weight of the pen and affects users' experience. Moreover, it is mainly used to achieve interaction with the device rather than to correct the pen-holding posture in writing. Therefore, none of the existing works are readily available and focus on detecting pen-holding gesture.

Smartwatches are worn on the wrist and can run a variety of applications. It is often equipped with numerous built-in sensors, and can communicate wirelessly with nearby devices. The pen-holding gesture detection based on the smartwatch can provide a prompt function to real-time monitor the pen-holding gesture. People only need to install an app on their smartwatches to effectively recognize the pen-holding gesture, making it more convenient and easy to use. Recent research also reveals that the smartwatches are becoming more and more popular. It is estimated that smartwatch

The associate editor coordinating the review of this manuscript and approving it for publication was Alba Amato.

volumes will reach a total of 46.2 million units shipped in 2018, up to 38.9% from the 33.3 million units shipped in 2017 [7]. These favorable conditions overcome the shortcomings of previous technologies. So it is valuable to develop a pen-holding gesture recognition system based on smartwatch.

In this paper, we present the design, implementation and evaluation of SmartGe, which can detect pen-holding gesture leveraging built-in sensors of smartwatches when writing. Considering the huge number of English words and Chinese characters (more than 5,000 commonly used), it is difficult to construct a classification system based on each word or character to recognize the pen-holding gesture. Fortunately, all words are composed of 26 letters. Similarly, all Chinese characters are composed of 32 strokes. For English words, we can detect the gesture by extracting any of the 26 letters written. However, due to complex two-dimensional structure, the same stroke signal may produce different signals in different locations of the Chinese character. The accuracy of detecting the pen-holding gesture based on 32 strokes is very low. After further study and observation, we found that more than 90% of Chinese characters contain the horizontal and vertical strokes. And a novel technique of combined strokes to address this difficulty is proposed. Recognizing different pen-holding gestures based on the combination of these two kinds of strokes is practicable.

A fully functional prototype of SmartGe has been implemented, and extensive experiments with 12 volunteers have been conducted for performance evaluating. The handwriting detection algorithm is designed to extract stroke or letter. Then we use the data augmentation technology and DCNN technology to construct the first classifier to distinguish strokes and letters. Specially, after detecting strokes, we assembles the signals of one vertical stroke and one horizontal stroke head-to-tail in sequence as input for next step classification (the order of horizontal stroke and vertical strokes is arbitrary). Finally, we construct the second classifier to recognize the pen-holding gestures based on the combination strokes and letters. According to the experimental results, SmartGe can identify pen-holding gesture as up to 98.3%. It's very useful for people to provide a lightweight system for monitoring the writing quality and habits continuously. With the design of SmartGe, we make the following key contributions.

- We propose an SmartGe model based on commodity smartwatches, which can detect pen-holding gestures on the paper.
- We design a handwriting detection algorithm to avoid incomplete or redundant signal detection, which can correctly shape the window that contains the complete signal of stroke/letter.
- With limited training samples, we use data augmentation to avoid over-fitting, which can improve the performance compared with none data augmentation processing. And we realize the model using the DCNN method and conduct extensive experiments to evaluate the performance of SmartGe. The average recognition accuracy of pen-holding gestures is 98.3%, which validates the effectiveness and robustness of SmartGe.

The rest of the paper is organized as follows. In Section II, we briefly introduce the related works. We present the overview of the architecture of SmartGe in Section III. The design and architecture of SmartGe with detailed descriptions in Section IV. Experimental results are given in Section V, and we discuss the limitations and summarize our work in Section VI.

II. RELATED WORK

A. INERTIAL SENSOR BASED WRITING RECOGNITION

Some wearable devices, such as smartpen and smartwatch, can be used to detect handwriting. In [8], a system is proposed that six surface EMG sensors are attached to the forearm for handwriting recognition. In [9], a sensor attached on the fingertips can be used to recognize 36 handwriting vocabularies including 10 digits and 26 English lower-case characters. In [10], a system shows the danger of handwriting content leakage from smartwatches' motion sensors by recording the motions and extracting handwriting-specific features. Previous handwriting gesture recognition systems utilizing inertial sensors focus on simple text recognition, such as English letters and numbers. However, there is no existing work based on smartwatch to recognize the pen-holding gesture.

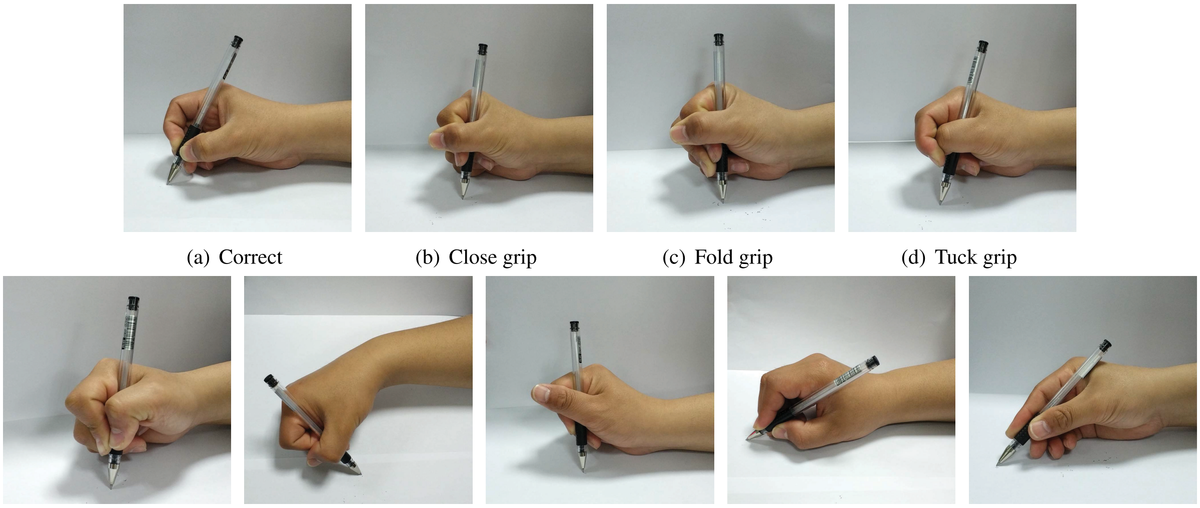
B. WIRELESS SIGNAL BASED WRITING RECOGNITION

Due to the pervasiveness of Wi-Fi access points, Wi-Fi signals have been widely exploited for recognizing gestures and tracking motions. In [11], the Wi-Wri system utilizes Channel State Information (CSI) extracted from Wi-Fi signals reflected from the hand movement to identify 26 English Letters gestures of mid-air handwriting. In [12], the Writing-Hacker system is designed to recognize handwritten English words based on acoustic signals and acceleration signals in mobile devices. Wireless signals, though having no visual range limit and can bypass obstacles, are easily interfered by other signals on the same unlicensed band, which significantly affects the performance of the system.

C. VISUAL BASED WRITING RECOGNITION

Other existing pen-holding gesture system using image processing techniques is also sensitive to lighting conditions [13]. Moreover, the computation complexity of imaging processing is usually high. And vision-based systems require augmentation of the environment to identify pen-holding gesture using specialized sensing devices, which adds cost to their deployment.

Our proposed SmartGe system proves that people can leverage the inertial sensors built-in smartwatch to recognize pen-holding gesture, which is robust to lighting conditions and can avoid interference of wireless signals compared with existing works. It is convenient for users just to wear a smartwatch and implant an app, which is a common habit for most



(a) Correct (b) Close grip (c) Fold grip (d) Tuck grip

(e) Squeeze grip (f) Hook grip (g) Wrap grip (h) Mount grip (i) Tripod grip

FIGURE 1. Correct and incorrect pen-holding gestures.

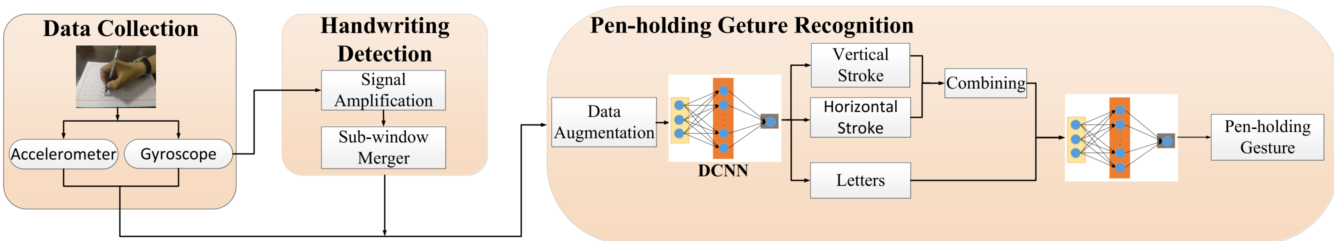


FIGURE 2. SmartGe system overview.

users and makes it promising for both academic research and industrial application.

III. OVERVIEW

In this section, we will outline the system design of SmartGe. The system is designed for recognizing pen-holding gesture in both English and Chinese writing only relying on the smartwatch, which can distinguish 9 different types of pen-holding gestures, including one correct and eight incorrect [14]. Fig. 2 illustrates the architecture of SmartGe.

- *Data Collection.* We collect data from the accelerometer and the gyroscope, the two most common built-in sensors in the smartwatch, with a sampling rate of 100Hz.
- *Handwriting Detection.* To build a more accurate classification model for recognizing handwriting gestures, we develop a handwriting detection algorithm to identify the event of writing by using the signals of strokes or letters as input to avoid detecting incomplete or redundant signals.
- *Pen-holding Gesture Recognition.* To enhance model generalization and improve accuracy, we increase sample size using data augmentation method for extracted signals. We construct two-level serial classifiers to recognize the pen-holding gesture. The first level is used to distinguish letters in English and strokes in Chinese. The detected handwriting data (angular velocity and acceleration) serves as the input of the first level classifier, and set the output labels as letters, horizontal strokes, vertical strokes and other strokes. Then we connect the data of a horizontal stroke and a vertical stroke in an arbitrary order. We set the merging strokes and letters as input to the second level classifier for pen-holding gesture recognition, which can realize the fusion recognition of strokes and letters. Two classifiers are both trained using DCNN. In conclusion, if the handwriting signal is recognized as a letter, the pen-holding gesture can be detected directly. If the signal belongs to horizontal stroke or vertical stroke, the two signals are retained and combined for the pen-holding gesture detection. If the signal belongs to other strokes, the signal will be abandoned. Therefore, SmartGe is independent of character and word independence. That is to say, no matter what characters or words we write, the pen-holding gesture can be detected by us normally.

IV. SMARTGE: DESIGN DETAILS

In this section, the detailed design of SmartGe is presented. We describe the process of writing detection, based on which the pen-holding gesture recognition model is build.

A. HANDWRITING DETECTION

Before performing the pen-holding gesture recognizition, we need to extract the signal of a stroke or letter. As shown in Fig. 3(a)(b), the influence of handwriting gesture on angular velocity is more obvious than acceleration, so we design handwriting detection algorithm based only on the angular velocity.

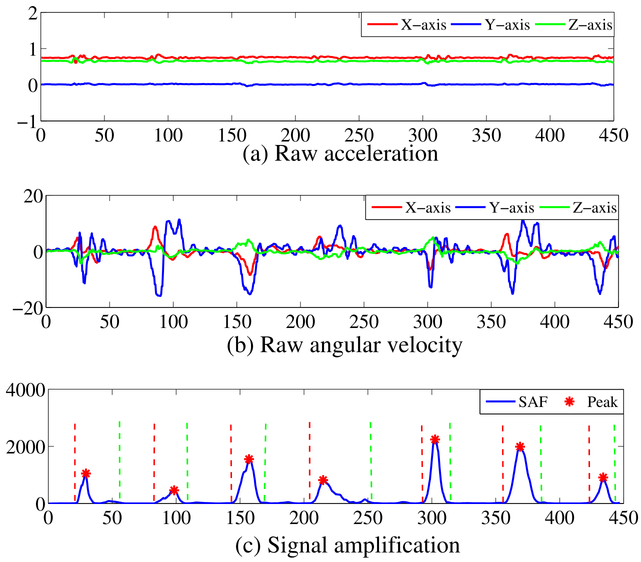


FIGURE 3. Signal of stroke/letter.

1) SIGNAL AMPLIFICATION

Due to the hand-shaking and internal noises of sensors, some signals with low energy are easily confused with random noise. To increase the signal-to-noise ratio (SNR), the signal of angular velocity is conversed to time-frequency domain by wavelet de-noising method [15]. We also design a signal amplification method to increase the SNR further. Morever, the wrist will slightly lift up and then fall down to complete a stroke/letter, which can be captured by the z-axis of gyroscope. Therefore, we amplify the signal of z-axis and then find the handwriting signal accurately through the peak detection. This can avoid the phenomenon that the larger noise is detected as pen-holding gesture only through the window function.

$$SAF[i] = (\sum_{j=i-5}^{i-1} |z_j| + \sum_{j=i+1}^{i+6} |z_j|).^2. \tag{1}$$

where $SAF[i]$ is signal amplification function of the z axis of angular velocity z_i. We use the signal with a macro frame size of 12 samples (120ms) for each term in the product and produce high magnitude peaks at shot impact points [16]. The Fig. 3(c) shows the calculation result. The magnitude changes of the gyroscope after signal amplification are much larger than the signal of noise. Therefore, we can find the handwriting signals more effectively with peak detection.

2) SUB-WINDOW MERGER ALGORITHM

After locating the pen-holding gesture, we define the boundary of the amplified signal with the window function for covering with the complete pen-holding gesture. The traditional fixed sliding window usually judges whether the signal energy within the window is larger than the threshold (the average energy of random noise) to determine whether the event occurs, such as handwriting events [17]. Considering different people's writing habits, the writing time of strokes and letters is different and the choice of window size is essential. A large window may cover redundant noise signals while a small window may extract incomplete event signals. Therefore, we need to design a novel sliding sub-window merging algorithm for solving this problem. The main idea is to detect signals by using small-sized sub-windows and then merge these sub-windows into a parent window that the complete signals should be contained. We set 50% overlapping rate for a window of N sampling points. Considering that the amplified signal has a higher SNR, it is more obvious to distinguish the noise by window detection, and the amplified signal does not change the boundary of the acceleration and angular velocity signals. So we determine the stroke/letter boundaries with the help of SAF signals and calculate the energy of the SAF signals in sub-window I as follows.

$$Energy_I = \sum_{i \in I} (SAF[i]^2), \tag{2}$$

The $Energy_I$ represents the energy values of the amplified signal and $i \in I$ reflects data at a point of SAF in sub-window I. We use $Energy_I$ values to determine the boundary of the parent window. Every time the $Energy_I$ of a sub-window is lagerer than the threshold of noise energy, we consider that writing initiation has completed and the signals in the sub-window are kept as the beginning of parent window. When the signal duration exceeds 0.4 seconds, we no longer monitor the sub-window signal[1] and we regard the last sub-window as the end boundary of parent window. Then we save the first $N/2$ sampling points in each sub-window with the overlapping rate of 50% [14].

$$D = D \cup \{SAF_1^I, SAF_2^I, \cdots, SAF_{N/2}^I\}. \tag{3}$$

A parent window D, which contains the complete SAF signal, consists of the sub-windows from beginning to end. The two sides boundary of SAF signal are also the signal boundary of one letter/stroke. Therefore, both signal of acceleration and angular velocity of the letter/stroke are saved within the parent window. The problem detecting incomplete or redundant signals can be avoided by the proposed sliding sub-window merging algorithm shown in Alg. 1, which contributes to more accurate classification performance.

B. PEN-HOLDING GESTURE RECOGNITION

After the signal detection is completed, we perform the following two processes to identify the pen-holding gesture.

[1]The response time of a normal person is about between 0.15 second and 0.4 second [18].

Algorithm 1 Handwriting Detection Algorithm

Require: z_i. // The value of the z axis from the gyroscope.
Ensure: D. // The parent window.

SAF calculation:

$$SAF[i] = (\sum_{j=i-5}^{i-1} |z_j| + \sum_{j=i+1}^{i+6} |z_j|).^2. \tag{4}$$

Setting the sub-window size as N sampling points, and the overlapping rate as 50%.
$t_0 := 50\%$ Time duration of a sub-window.
$I = 0$. // Index of the current sub-window.
$flag = 0$. // No writing event.
while The detection module is active **do**
  $I = I + 1$.
  The energy calculation of SAF signals in sub-window i:

$$Energy_I = \sum_{i \in I} (SAF_i^2).$$

  if $Energy_I >$ threshold&$flag = 0$ **then**
    $flag = 1$. // The start of the boundary detection.
    $SAF = \Phi$.
  else
    if $Energy_I <$ threshold &$flag = 1$ **then**
      if $Energy_{I-1} >$ threshold **then**
        $t = t_0$. // The possible end of the boundary.
      else
        $t = t + t_0$.
      end if
      if $t > 0.4$ **then**
        $flag = 0$. // The definite end of the boundary.
      end if
    end if
  end if
  if $flag = 1$ **then**
    Saving the first $N/2$ sampling points in sub-window I:

$$D = D \cup \{SAF_1^I, SAF_2^I, \cdots, SAF_{N/2}^I\}.$$

  end if
end while

Firstly, we build the first classifier to differentiate strokes and letters and then combine one horizontal stroke and one vertical stroke without order. Secondly, the pen-holding gesture recognition model can be build based on the combined strokes and letters. We use the traditional classification method based on generic features and the deep learning respectively to train classification model, which are described in detail in the following sections.

1) DATA AUGMENTATION

To overcome the deficiency of training data, we use data augmentation method on raw data of acceleration and angular velocity to enlarge our training dataset, which can avoid over-fitting and improve recognition accuracy effectively. Since the variation of writing habits and deviation when placing the sensor result in different patterns of the signal, we adopt several methods to stimulate the variations to cover the undetected input space. For example, time-warping is one way to perturb the temporal location. By distorting the time intervals of the samples smoothly, we stimulate the changing pace among strokes when writing a character. Magnitude-warping changes the magnitude of each sample randomly around one, which stimulates the differences of strength of strokes or letters. Scaling changes the magnitude of all data in a sample by multiplying by a random scalar. Therefore, we transform the data using time-wraping, magnitude-warping and scaling respectively and add these new datasets to our sample for enhancing the robustness of our model and ensuring a good generalization ability [19].

2) CONSTRUCTING CLASSIFICATION MODEL WITH DCNN

a: DATA MAPPING

We transform the time-series data of six-axis sensors (including gravitational acceleration and gyroscope angular velocity) to a feature map with $H \times W$ tensor as input to the DCNN network, where H represents the signal length in the time domain, W represents the number of axes ($H = 6$) [20], [21]. Since different strokes or letters have different signal lengths in the parent window, i.e., the size of W is not uniform, we need to unify the size of the feature map for the input layer in a suitable way for subsequent convolutional processing. We set the signal length W as 320 points, because most strokes or letters can be written within 3.2 seconds. Longer signals will be truncated equally at the front and the end, and shorter signals will be patched with zeros at the end. We normalize the six-axis data using deviation standardization to eliminate the influence of dimensions on handwriting recognition.

b: ARCHITECTURE OF DCNN

The structure of the DCNN used in this paper is shown in Fig. 4, which consists of an input layer, two convolutional layers, each followed by a Relus layer, two maximum pool layers and a fully connected layer followed by a softmax layer that outputs the probability for each class. All labels are coded using one-hot method [21], [22].

The input to our DCNN has a fixed size of 320×6. Each convolutional layer has 800 convolutional kernels. The parameters of each convolutional kernel are optimized by the backpropagation algorithm, the purpose of which is to extract different characteristics of the input. Each convolutional layer has a kernel size of 16 samples with a moving step of 1. The output of the convolutional layer is the input to the max-pooling layer that selects the maximum value of each feature map to reduce dimension and avoid over-fitting. Each max-pooling layer uses a moving pooling window with a length of 16 samples and a moving step of 4 for dimensionality reduction. All feature maps are automatically zero-padded

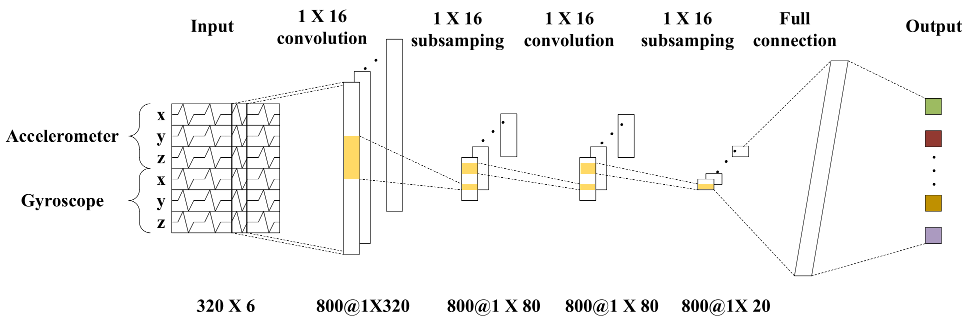


FIGURE 4. Architecture of DCNN framework. (The number of feature maps and the dimensions of a feature map in each layer is shown before and after "@".)

after the layer. The output of the second max-pooling layer is transformed into 1-dimensional data by flattening, as the input to the last fully connected layer. The fully connected layer uses the dropout method with a probability ($p = 0.5$) to avoid over-fitting. To obtain a probabilistic prediction, the softmax method is used in the last layer to get the probability distribution of each class. We optimize the parameters using the root mean square loss and the Adam Gradient Descent method. The learning rate is set as 0.0001.

3) CONSTRUCTING CLASSIFICATION MODEL WITH GENERIC FEATURES

a: FEATURE EXTRACTION

We extract features from the time domain, the frequency domain, and the time-frequency domain, in order to cover a wide range of relevant features across different frequency and time resolutions. In the time domain, we calculate the cross-correlation coefficients between each pair of axes of each sensor. We also derive typical temporal statistics including mean, variance, standard deviation (STD), maximum energy in the window (E^i_{max}), minimum energy in the window (E^i_{min}), autoregressive, kurtosis, skewness and deviation, which can characterize the unique temporal patterns of strokes. We also add the number of peaks obtained through a peak detection algorithm as our another feature. We transform the signals from the time domain to the frequency domain through Fast Fourier Transform (FFT), obtaining the frequency components, frequency distribution range, and the energy in every frequency component. Since different strokes generate different frequency distributions and the stroke signals mainly concentrate on low frequencies, we select the first 20 FFT coefficients as features [23]. We also add frequency domain statistics including median, energy and variance to the feature set. We attain features in the time-frequency domain through the Discrete Wavelet Transform (DWT). We compute low-frequency coefficients and high-frequency coefficients at different scales using the Daubechies wavelet decomposition as the time-frequency features. We also add time-frequency domain statistics including the mean square root, variance and the mean of the wavelet coefficients to our feature set. Finally, we have extracted 728 features from the three axes of both the accelerometer and the gyroscope data, which can capture the characteristics of unique gesture-related patterns.

b: CLASSIFICATION

With the features learned above, we use a supervised learning method to train the model. Finally, we build a stroke-direction classifier for distinguishing the stroke and letter. After identifying these two types of signals, a pen-holding classifier will identify the pen-holding types.

V. EVALUATION

A. DATA COLLECTION

We recruit 12 volunteers with HUAWEI Watch on their right wrists, including 8 males and 4 females aged 19 $\sim$ 26. An in-depth analysis is performed by recording the video of all experiments. We ask volunteers to write 8 Chinese characters include horizontal stroke and vertical stroke randomly and two English sentences: "the quick brown fox jumps over the lazy dog" and "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG" that contain all lower-case English letters and upper-case English letters, each character or word for 20 times, with 9 kinds of pen-holding gestures (1 correct gesture and 8 incorrect gestures) respectively. In total, we have collected $17,280(12*9*8*20)$ Chinese characters and $38,880(12*9*18*20)$ words for pen-holding gesture detection.

B. PARAMETER SELECTION

As mentioned above, we propose the handwriting detection algorithm to identify signals of letters/strokes. The signal amplification method is used to detect handwriting gesture signal and the sub-window merger algorithm is used to determine the signal boundary. Considering that the size of the sub-window directly affects the accuracy of pen-holding gesture recognition, it is necessary to choose a suitable sub-window size. We evaluate relative detection errors under different sub-window sizes. We define S_d as the number of detected signals while S_t represents the real number of signals. The relative error can be defined as follows.

$$\delta = \frac{|S_d - S_t|}{S_t}. \tag{5}$$

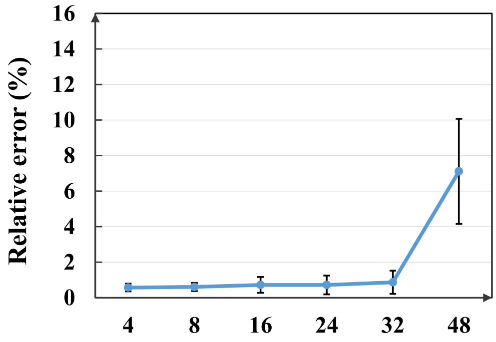


FIGURE 5. Relative error with different sub-window sizes.

Fig. 5 shows that the relative error maintains stability in a relatively low error range as the sub-window size increases to 32. And we can find that when the sub-window size is 4 (0.04s), the detection performance is the best with the standard deviation of 0.56%. The sub-window size can be set as 4 in all following experiments. When the sub-window size is larger than 32 (0.32s), the relative error increases evidently. The reason for this phenomenon is that the large sub-window would merge multiple adjacent gesture signals into the same parent window.

C. PEN-HOLDING GESTURE RECOGNITION

In this section, we introduce the evaluation metrics of the system firstly. Then we compare the effects of different classification models and different combinations of strokes and letters on the recognition accuracy. Finally, we verify the robustness of the system.

1) EVALUATION METRICS

We randomly divide the dataset into the training set, the validation set and the test set according to the 80-10-10 split principle to avoid introducing bias. We train the classifier using the training set, tune the parameters using the validation set and evaluate the performance using the test set. The precision, recall and F1-score is used for the performance evaluating from different aspects [14], [24].

2) RECOGNITION PERFORMANCE

a: PARAMETERS CONFIGURATION

Four different machine learning classifiers, i.e., random forest (RF), support vector machine (SVM), k-nearest neighbor (KNN) and DCNN are used for comparison. The optimal values of the parameters for SVM (parameter *cost* and γ) and KNN (parameter K) are determined via grid search through the range $cost \in \{1, 10, 100, 1000\}$, $\gamma \in \{0.0001, 0.01, 0.1, 1\}$ and $K \in \{1, 2, 3, 4, 5\}$. For RF, grid search is performed on the number of iterations ($n\_estimators \in \{10, 20, 30, 40, 50\}$), the maximum depth of the decision tree ($max\_depth \in \{1, 2, 3, 4, 5\}$) and the minimum number of samples ($min\_samples\_split \in \{50, 70, 90, 110, 130\}$) to find the optimal parameter values. As a result, for SVM, the optimal *cost* and γ are 1000 and 0.1 respectively; for KNN, the optimal K is 5; the optimal *n_estimators*, *max_depth* and *min_samples_split* are 30, 2 and 50 respectively [21].

For DCNN, we observe that the size of the kernel & pooling and the number of layers have an important influence on the recognition performance. As shown in Fig. 7, when the kernel & pooling size is 16, the recognition performance is the best. The DCNN's two-layer model (one convolutional layer and one pooling layer), the four-layer model and the six-layer model have recognition rates of 92.4%, 98.3% and 98.6% respectively. The results show that with a larger kernel & pooling size and more layers, more meaningful information can be learned. Considering the training cost and energy consumption, we finally choose the four-layer model (2 convolutional layers and 2 pooling layers) with a kernel & pooling size of 16 [25].

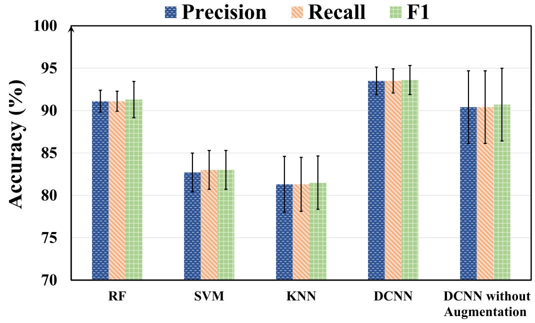
b: STROKE/LETTER CLASSIFICATION MODEL

The trials with four types of classifiers, i.e., random forest (RF), support vector machine (SVM), k-nearest neighbor (KNN) and DCNN are conducted and compared. The first layer classifier is used to distinguish letters, vertical strokes, horizontal strokes and other strokes. As shown in Fig. 6(a), the recognition precision, recall and F1-score of DCNN are 93.1%,93.1% and 93.2% respectively, which has the maximum recognition rate. Therefore, we build the classifier using the DCNN. We also evaluate the performance of the DCNN model without data augmentation, the mean recognition accuracy of which is about 91% and is about 2% lower than that with data augmentation. It is proved that after data augmentation, including magnitude-warping, scaling and time warping, our dataset can cover some of the undected input space and improve the performance of the model.

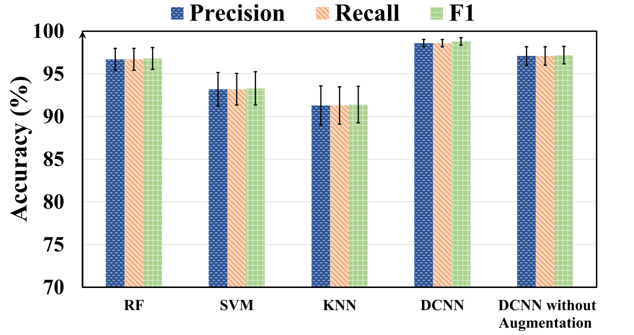
c: PEN-HOLDING GESTURE CLASSIFICATION MODEL

Based on the collected letters and strokes, we construct the second layer classification model for pen-holding gesture recognition. Similarly, we compare the performance of these four classifiers. From the Fig. 6(b) we can see, the DCNN with augmentation has the best performance with the recognition precision, recall and F1-score being 98.2%, 98.2% and 98.3% respectively. The recognition accuracy without data augmentation is about 97%, which is lower than the accuracy with data agumentation. This further evidences that the recognition performance of our system can be improved using data augmentation method.

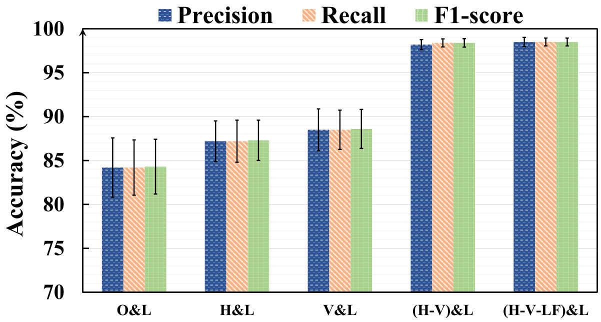
Fig. 6(c) shows the recognition accuracy with different strokes and letters. The detection accuracies relying on the horizontal stroke and letter (H&L) and the vertical stroke and letter (V&L) are 87% and 89% respectively, while the connected stroke of horizontal and vertical and letter ((H-V)&L)) has a much higher accuracy of 98.3%. We also add the left-falling stroke (LF), a common stroke in many Chinese characters, to connected stroke for comparison. The recognition accuracy of ((H-V-LF)&L) is up to 98.5%, where



(a) Accuracy of stroke/letter recognition with different classifiers.

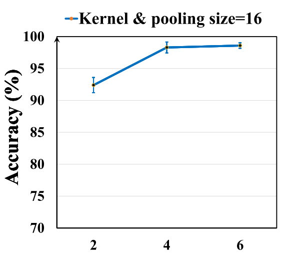


(b) Accuracy of pen-holding gesture recognition with different classifiers.

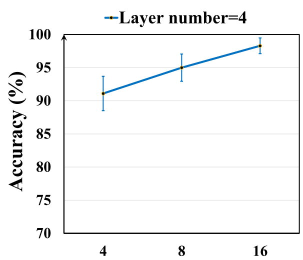


(c) Accuracy of pen-holding gesture recognition with different combinations of strokes and letters.

FIGURE 6. Performance evaluation of the model.



(a) Number of layers



(b) Kernel & pooling size

FIGURE 7. Accuracy with different parameters configuration of DCNN.

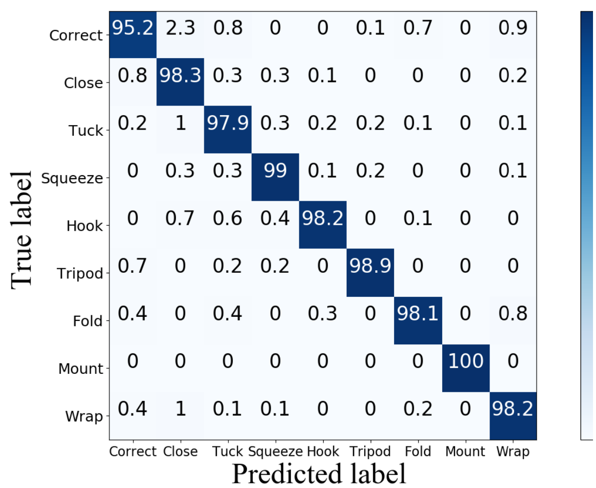
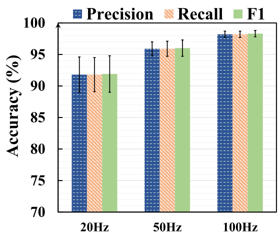


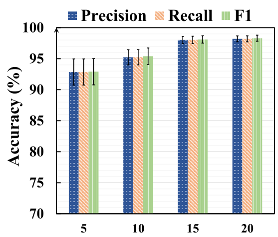
FIGURE 8. Confusion matrix of 9 kinds of pen-holding gestures.

the improvement is not very significant. All other strokes and letters (O&L) have a much less presence and may even degrade the recognition accuracy as low as 84.3%. The possible reason is that some different complicated strokes may exhibit consistent patterns when writing with different pen-holding gestures. Therefore, three types of data include the combination of horizontal stroke and vertical stroke, and the letters can be used to improve the performance of the system.

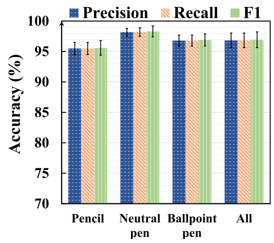
Fig. 8 shows the confusion matrix of the 9 types of pen-holding gestures. It shows that the correct gesture is misclassified as the close grip gesture with a probability of 2.3%. And the mount grip gesture can be recognized with a possibility of 100%. From Fig. 1 we can see, the correct gesture and close grip gesture are similar, which makes it's hard to distinguish accurately. The back of the hand of the mount grip gesture is upward, different with other gestures, which can be identified precisely. To summarise, the accuracy of misclassification of all pen-holding gestures is very low, which can help us to recognize the pen-holding gesture well.



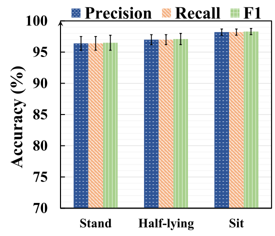
(a) Sampling frequency



(b) Training size



(c) Pen type



(d) Writing posture

FIGURE 9. Different influence factors of the model.

3) ROBUSTNESS

To verify the effectiveness and robustness of the model, we have asked the same 12 participants to conduct each experiment to evaluate the factors from different aspects that influence model, including sampling frequency, training size, pen type, writing posture and wearing method.

a: IMPACT OF SAMPLING FREQUENCY

The influence of different sampling frequencies (20Hz, 50Hz,100Hz) on the recognition accuracy of pen-holding gestures for all participants are also compared in detail. It can be seen from Fig. 9(a) that with the increase of sampling frequency, the recognition accuracy of pen-holding gesture has a slight improvement. The accuracy is 91.8% with 20Hz of sampling frequency while the accuracy is up to 98.3% with 100Hz of sampling frequency. Therefore, the higher the frequency, the richer the captured information is, the better the recognition performance is. However, more energy is lost.

b: IMPACT OF TRAINING SIZES

We also evaluate the SmartGe under different training sizes [26]. As shown in Fig. 9(b), with the training instances increasing from 5 to 20, the performance improves gradually. When there are 5 instances per pen-holding gesture, the detection accuracy is 92.8% while 20 instances is up to 98.3%. Therefore, the user experience will be improved with the accumulation of personal data. It also shows that SmartGe has a relatively good performance with limited training data.

c: IMPACT OF PEN TYPES

Considering that different types of pens may affect the recognition performance of the pen-holding gesture, we find three types of pens to carry out experiments and compare their recognition accuracy, including a diameter of 10mm, a gel-ink pen with a diameter of 8.5mm and a pencil with a diameter of 7mm. As shown in Fig. 9(c), the recognition accuracies of all types of pens are over 95%. It confirms that SmartGe is robust under different kinds of pens.

d: IMPACT OF WRITING POSTURES

We evaluate the performance of the model with different postures including stand, half-lying and sit in a chair respectively, which commonly occurs in the process of writing. As shown in Fig. 9(d), the mean accuracy when stand, half-lying and sit in a chair is 96.4%, 97% and 98.3% respectively, which verifies that our system is robust with different postures.

TABLE 1. Impact of wearing method.

Tightness&Distance	Mean	Standard deviation
Tight&Near	98.3%	0.5%
Tight&Far	96.9%	0.9%
Loose&Near	97.9%	0.6%
Loose&Far	96.0%	1.1%

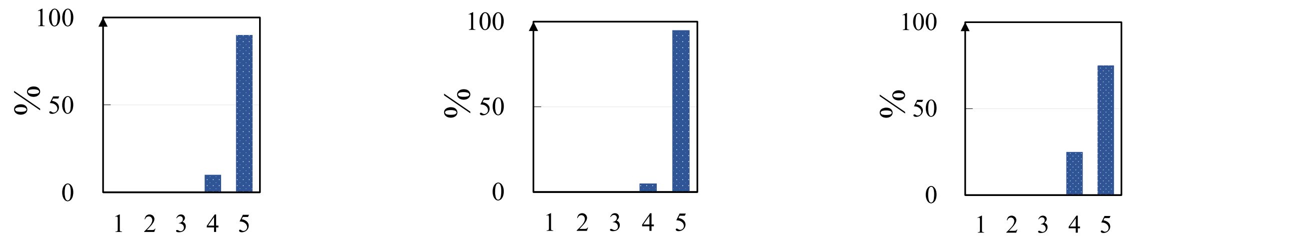
e: IMPACT OF WEARING METHOD

We also evaluate the robustness based on different wearing methods, including tightness and distance from wrist. When the smartwatch is worn loosely/tightly, the random noise will occur. Similarly, when the smartwatch is far from wrist, it may lose some gesture information. The tightness of the system is evaluated in two way: Loose(circumference of 17.6 cm) and Tight(circumference of 16.2 cm), where perimete of the selected user's wrist is 16 cm. We also evaluate the influence of distance from smartwatch to wrist, including Far distance (distance = 8cm) and Near distance (distance = 1cm). We ask all participants to wear a smartwatch according to our instructions to write with nine different pen-holding gestures respectively. And the extracted data is used to train and evaluate. Table. 1 shows the average recognition accuracy on four different combinations of methods. The tighter the wear is and the closer the distance is, the higher the recognition accuracy of the pen-holding gesture is. Nevertheless, the recognition accuracy of the pen-holding gesture can achieve good performance in different wearing methods. It shows that our system have robustness to wearing method [25].

D. DELAY AND ENERGY CONSUMPTION

We evaluate the average delay of SmartGe. As listed in Table. 2, the average delay for the model is 116 ms, which indicates that the algorithm is effective and the delay of the system is satisfied.

We also analyze the power consumption of SmartGe using the battery drain rate ($\%/h$) of the smartwatch [27] for



(a) How do you like SmartGe? (b) How easy it is to use SmartGe? (c) How comfortable wear the smartwatch on the right?

FIGURE 10. User questionnaire.

TABLE 2. Delay and energy consumption.

	Mean	Standard deviation
Delay time of SmartGe	116 ms	6.5 ms
Power consumption with SmartGe	40%	3.6%
Power consumption with optimized SmartGe	28%	7.3%
Power consumption without SmartGe	4.2%	0.9%
Power consumption with Heart Rate Measuring App	45%	2.2%
Power consumption with Aller ID	26%	1.9%

all participants. The battery drain rate measures the average hourly decrease in battery levels. We compare the average battery drain rate with and without (standby mode) SmartGe running on the smartwatch for five hours. As shown in Table 2, the average battery drain rate with SmartGe is 40%, which implies that a fully-charged smartwatch can last about 2.5 hours for users to continuously use SmartGe. We compare the energy consumption of SmartGe with two other common functions of the smartwatch, i.e., heart rate measurement and vibration-based caller ID. We measure the battery drain rate of these two functions separately and continuously for five hours. As shown in Table 2, SmartGe has a comparable energy consumption to that of the heart rate application. Considering that users rarely write strokes or letters continuously for hours, we modify the code such that the SmartGe application sleeps when it detects no handwriting and wakes up when handwriting is detected. The average battery drain rate of the optimized SmartGe is 28%, which is similar to that of the vibration-based caller ID.

In conclusion, smartwatch-based SmartGe can realize an ideal recognition performance. Therefore, the SmartGe may be used to assist people in writing.

E. USER FRIENDLINESS

We also ask these participants to answer our questionnaires on their experience with SmartGe. Each participant wears a smartwatch and writes for thirty minutes. Based on the scale of 1 to 5 corresponding to 'Very dislike/ difficult/ uncomfortable' to 'Very like/easy/comfortable', the results are showed in Fig. 10(a)(b)(c) confirm that users strongly support us the SmartGe system. All participants agree that the they are comfortable with the smartwatch on the right for pen-holding gesture recognition. This helps us further improve our design of SmartGe.

In conclusion, smartwatch-based SmartGe can realize an ideal recognition performance, which may be used to assist people in pen-holding gesture recognition.

VI. LIMITATIONS AND CONCLUSION

A. LIMITATIONS

The number of volunteers and the number of samples from each volunteer are relatively small due to time and budget limitations, more volunteers can be recruited and more data can be collected to improve the performance of SmartGe. Our experiments only involve right-handed users, and it is interesting to see how SmartGe performs on left-handed users or even ambidextrous users. Moreover, the SmartGe needs to write slowly with stroke by stroke. Therefore, our system is more suitable for beginners in Chinese such as young kids. For other people who write cursives, the SmartGe doesn't work well. And we will further study it in the future.

B. CONCLUSION

In this paper, we present SmartGe, a commercially available system based on smartwatch, which can identify pen-holding gesture in both English and Chinese. The result shows an average accuracy of 98.3%, which is high enough to recognize the pen-holding gesture. Extensively experiments also confirm that the SmartGe system is effectiveness and robustness. As SmartGe is well portable and easy-to-use, it will become an ideal tool to improve the habits and quality of handwriting, especially for early education.

REFERENCES

[1] (2018). [Online]. Available: http://www.sohu.com/a/241820740_497957

[2] (2018). *How to Hold a Pen*. [Online]. Available: https://www.wikihow.com/Hold-a-Pen

[3] A. Murata and K. Gotoh, "Effects of pen holding posture on handwriting motion," in *Proc. Adv. Social Org. Factors*, vol. 12 2014, p. 469.

[4] H. Schwellnus, H. Carnahan, A. Kushki, H. Polatajko, C. Missiuna, and T. Chau, "Effect of pencil grasp on the speed and legibility of handwriting in children," *Amer. J. Occupational Therapy*, vol. 66, no. 6, pp. 718–726, Nov. 2012.

[5] (2012). *Writing Instrument Gripping Aid and Writing Instrument Having the Same*. [Online]. Available: https://patents.google.com/patent/US8926203B1/en

[6] H. Song, H. Benko, F. Guimbretiere, S. Izadi, X. Cao, and K. Hinckley, "Grips and gestures on a multi-touch pen," in *Proc. Annu. Conf. Human Factors Comput. Syst. (CHI)*, 2011, pp. 1323–1332.

[7] U. Jitesh, L. Ramon, and S. Michael. (2018). *IDC Forecasts Slower Growth for Wearables in 2018 Before Ramping Up Again Through 2022*. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=prUS44276818

[8] S. Tan, Y. An, Y. Wu, and D. Zhang, "Electromyography based handwriting recognition system using LM-BP neural network," in *Proc. 9th Int. Conf. Hum. Syst. Interact. (HSI)*, Jul. 2016, pp. 83–88.

[9] L. Jing, Z. Dai, and Y. Zhou, "Wearable handwriting recognition with an inertial sensor on a finger nail," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1330–1337.

[10] Q. Xia, F. Hong, Y. Feng, and Z. Guo, "MotionHacker: Motion sensor based eavesdropping on handwriting via smartwatch," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2018, pp. 468–473.

[11] X. Cao, B. Chen, and Y. Zhao, "Wi-Wri: Fine-grained writing recognition using Wi-Fi signals," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1366–1373.

[12] T. Yu, H. Jin, and K. Nahrstedt, "WritingHacker: Audio based eavesdropping of handwriting via mobile devices," in *Proc. Int. Joint Conf. Pervas. Ubiquitous Comput.*, 2016, pp. 463–473.

[13] H.-Y. Cheng, C.-C. Yu, V. Gau, and C.-L. Lin, "Video-based signature verification and pen-grasping posture analysis for user-dependent identification authentication," *IET Comput. Vis.*, vol. 6, no. 5, pp. 388–396, Sep. 2012.

[14] J. Zhang, H. Bi, Y. Chen, J. Chen, and Z. Wei, "SmartWriting: Pen-holding gesture recognition with smartwatch," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[15] S. Hadji, M. Salleh, M. Rohani, and M. Kamat, "Wavelet-based Performance in Denoising ECG Signal," in *Proc. 8th Int. Conf. Signal Process. Syst. (ICSPS)*, 2016, pp. 148–153.

[16] M. Sharma, A. Anand, R. Srivastava, and L. Kaligounder, "Wearable audio and IMU based shot detection in racquet sports," May 2018,*arXiv:1805.05456*. [Online]. Available: https://arxiv.org/abs/1805.05456

[17] J. Bandy, J. Knighten, and J. Payton, "Demonstrating highfivelive: A mobile application for recognizing symbolic gestures," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops*, Mar. 2016, pp. 1–3.

[18] Y. He, Y. Wang, L. Zhang, Q. Liu, and X. Wang, *The Design and Implementation of Intelligent Electronic Timer*. 2015, pp. 151–154.

[19] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson' disease monitoring using convolutional neural networks," in *Proc. 19th ACM Int. Conf. Multimodal Interact. (ICMI)*, 2017, pp. 216–220.

[20] H. Brock, Y. Ohgi, and J. Lee, "Learning to judge like a human: Convolutional networks for classification of ski jumping errors," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, 2017, pp. 106–113.

[21] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, and B. M. Eskofier, "Activity recognition in beach volleyball using a deep convolutional neural network," *Data Mining Knowl. Discovey*, vol. 31, no. 6, pp. 1678–1705, Nov. 2017.

[22] J. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.

[23] F. Kerber, P. Schardt, and M. Löchtefeld, "WristRotate: A personalized motion gesture delimiter for wrist-worn devices," in *Proc. 14th Int. Conf. Mobile Ubiquitous Multimedia*, 2015, pp. 218–222.

[24] D. Iyer, F. Mohammad, Y. Guo, E. Al Safadi, B. J. Smiley, Z. Liang, and N. K. Jain, "Generalized hand gesture recognition for wearable devices in IOT: Application and implementation challenges," in *Proc. Int. Conf. Mach. Learn. Data Mining Pattern Recognit.* Springer, 2016, pp. 346–355.

[25] J. Hou, X.-Y. Li, P. Zhu, Z. Wang, Y. Wang, J. Qian, and P. Yang, "SignSpeaker: A real-time, high-precision smartwatch-based sign language translator," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–15.

[26] C. Zhang, A. Bedri, G. Reyes, B. Bercik, O. T. Inan, T. E. Starner, and G. D. Abowd, "TapSkin: Recognizing on-skin input for smartwatches," in *Proc. ACM Interact. Surfaces Spaces (ISS)*, 2016, pp. 13–22.

[27] C. Min, S. Kang, C. Yoo, J. Cha, S. Choi, Y. Oh, and J. Song, "Exploring current practices for battery use and management of smartwatches," in *Proc. ACM Int. Symp. Wearable Comput. (ISWC)*, 2015, pp. 11–18.

HONGLIANG BI received the M.S. degree in electronic and communication engineering from the Soochow University of China, in 2015. He is currently pursuing the Ph.D. degree with the Computer School, Wuhan University. His research interests include the Internet of Things and smart sensing.

JIAN ZHANG received the B.S., M.S., and Ph.D. degrees in computer architecture from Wuhan University, in 1998, 2001, and 2006, respectively. He is currently a Professor with the Computer School, Wuhan University. His research interests include cloud computing, the Internet of Things, and big data.

YANJIAO CHEN (Member, IEEE) received the B.E. degree in electronic engineering from Tsinghua University, in 2010, and the Ph.D. degree in computer science and engineering from The Hong Kong University of Science and Technology, in 2015. She is currently a Professor with Wuhan University, China. Her research interests include spectrum management for Femtocell networks, network economics, network security, and Quality of Experience (QoE) of multimedia delivery/distribution.