

Article



Smartwatch User Interface Implementation Using CNN-Based Gesture Pattern Recognition

Min-Cheol Kwon ¹, Geonuk Park ² and Sunwoong Choi ^{2,*}

- ¹ Department of Secured Smart Electric Vehicle, Kookmin University, Seoul 20707, Korea; mincheol@kookmin.ac.kr
- ² School of Electrical Engineering, Kookmin University, Seoul 02707, Korea; zzaing0502@kookmin.ac.kr
- * Correspondence: schoi@kookmin.ac.kr; Tel.: +82-02-910-4416

Received: 25 July 2018; Accepted: 5 September 2018; Published: 7 September 2018



Abstract: In recent years, with an increase in the use of smartwatches among wearable devices, various applications for the device have been developed. However, the realization of a user interface is limited by the size and volume of the smartwatch. This study aims to propose a method to classify the user's gestures without the need of an additional input device to improve the user interface. The smartwatch is equipped with an accelerometer, which collects the data and learns and classifies the gesture pattern using a machine learning algorithm. By incorporating the convolution neural network (CNN) model, the proposed pattern recognition system has become more accurate than the existing model. The performance analysis results show that the proposed pattern recognition system can classify 10 gesture patterns at an accuracy rate of 97.3%.

Keywords: gesture pattern recognition; machine learning; smartwatch; Internet of things; wearable device; convolution neural network

1. Introduction

Recently, Internet of things (IoT) technologies that add the communication functionality to ordinary objects have been developed and are widely used [1,2]. Among these IoT devices, smartwatches are one of the most familiar and widely used devices. According to the Strategy Analytics, global smartwatch shipments increased by 34% YoY (Year-over-Year) in Q1 2018 [3]. The sales and usage of smartwatches are expected to increase in the future.

Along with an increase in the usage of smartwatches, many manufacturers are offering various smartwatch applications that meet the needs of various users. Users need various input methods to control these applications. Accordingly, the user interface for utilizing smartwatch applications is crucial, and thus, the user interface should be considered while developing a smartwatch and smartwatch applications [4,5].

Currently, most smartwatches use touchscreen and voice recognition as input methods. However, these input devices offer limited control of the user interface. A smartwatch is a device worn on the wrist, and its size and volume are compact. From the viewpoint of the touchscreen, it is difficult to install a touchscreen that is sufficiently large to offer various control functions. Such a limitation causes the user to touch at the wrong place, and user with a large hand would find it more difficult to control the interface. Second, voice recognition is currently used to write messages [6] or control simple functions with the help of a voice recognition secretary (Siri [7] or Bixby [8]). However, the voice recognition is used for natural language understanding and is not suitable for controlling a specific application. It is also less accurate when there is background noise and can distract others in a cubicle or classroom environment. Since it is difficult to install sensors or additional input devices on a

smartwatch due to the size limitation, an additional external input device that would allow for a communication with the smartwatch would be necessary for better control of the user interface [9].

Therefore, in this study, we aim to use an accelerometer installed on most smartwatches to recognize user gesture patterns, which are then used for the user interface. Application control methods using gesture patterns are already being used for various devices including smartphones and notebooks. For example, shaking a smartphone changes the music, and using two fingers up and down on a notebook controls the window scroll. Such control methods allow users to register convenient and often-used gesture patterns as a control command in the system, which can be used to control all the applications simply and conveniently.

We propose a method that classifies 10 gesture patterns using the convolution neural network (CNN) classification model. In the past, support vector machine (SVM) or dynamic time warping (DTW) methods have often been used. With SVM, the process by which specific features are extracted from raw data was crucial, but with CNN, raw data are used without extracting any features to prevent data loss that may occur in the feature extraction process. DTW classifies data by quantifying the similarity level between two sets of data; however, using CNN, a more accurate and refined analysis can be performed. The proposed CNN model shows a higher accuracy level than SVM or DTW by 8.1% and 1.3%, respectively.

This article is organized as follows. Section 2 describes the related research, and Section 3 shows the proposed system design. Section 4 describes the proposed gesture pattern sorter using CNN, and Section 5 evaluates the performance of the designed system. Section 6 presents the conclusion.

2. Related Works

This section surveys the research efforts on gesture pattern recognition systems based on the accelerometer for mobile devices. The objective of a gesture pattern recognition system is to classify the test gesture pattern set (that the user just performed) to a certain class according to the template gesture pattern set (that the user performed earlier).

Previous research can be mainly categorized into two types: the DTW based method [10] and the machine learning based method [11]. Intuitively, the DTW method measures the distance between the test gesture patterns dataset and the template gesture patterns dataset of each class and selects the class with the minimum distance [12–14]. Other research has used both the Hidden Markov Model (HMM) and DTW [15]. The HMM-based method is based on the probabilistic interpretation of gesture pattern samples to model the gestural temporal trajectory.

Machine learning uses techniques that give the computer an ability to learn an objective, which is based on big data, to achieve a specific goal. This is currently one of the most favored technologies and is widely used in gesture pattern recognition techniques. There are various algorithms in machine learning. Decision tree algorithm is an estimation model that uses a decision tree to connect the observed value and target value on an arbitrary item. Since its results are easy to understand and it does not need to process data, this algorithm is widely used in gesture pattern recognition [14,16]. The SVM algorithm finds the boundary that makes the largest margin between the two data. It allows for effective handling of complex data using a kennel [17]. SVM-based methods usually offer lower computational requirements at classification time, making them preferable for real-time applications on mobile devices. The gesture pattern classification accuracy depends closely on the feature vector for SVM-based methods. The SVM algorithm is extensively applied for motion gesture recognition [14,18,19]. It is used in gesture pattern recognition research where data are analyzed in the time or frequency domain or unnecessary data are separated based on a threshold [15,20]. Finally, the logistic regression algorithm is a probabilistic model that estimates the probability of the occurrence of an event using the linear connection of independent variables. Using the algorithm, six motions based on the gyroscope, accelerometer, and linear acceleration are used to improve the user interface [21].

It is important to note that we focus on the hand gesture pattern recognition system. It should be distinguished from the human activity recognition (HAR) system wherein whole body movement is considered. HAR using wearable devices has also been actively investigated for a wide range of applications, including healthcare, sports training, and daily activity recognition [22–25].

3. System Architecture

Figure 1 illustrates the overall architecture of our system, detailing the interdependencies between each piece of equipment. Our system consists of a smartwatch, smartphone, and server. The user runs a developed application on the smartwatch to initiate the system operation. More specifically, data is periodically collected and transmitted to the smartphone using Bluetooth. Once the smartphone collects the necessary number of samples, it transmits them to the server, which has a classifier to recognize the respective gesture pattern.



Figure 1. Overall system architecture.

3.1. Smartwatch

Amid the recent popularity of wrist-worn wearable devices like smartwatches or smart bands, many people use wearable devices. Most of them are equipped with an accelerometer. In this study, the Apple Watch Series 1, among the commercialized smartwatches, is used for our experiment. The Apple Watch Series 1 is equipped with a three-axis accelerometer and a Bluetooth communication module so it is very suitable for the test conducted in this study. The Apple Watch is operated using an iOS-based watchOS which allows for the development of applications through an SDK offered by Apple.

A smartwatch application is developed to collect data values of the acceleration on the x, y, and z-axes. Moreover, while performing a data capture, the user chooses a gesture pattern label from a list of available gesture patterns on the application before starting the gesture, as shown in Figure 2. This ensures that the sensing data are classified accordingly and that these labels are used for training the classifiers. Then, during a data capture operated by the start/stop button on the application interface, sensor data from the smartwatch are collected simultaneously and sent to a smartphone. After collecting the data of acceleration and gesture pattern label, the smartwatch transmits data to a smartphone using Bluetooth communication. In this study, data are sampled from the smartwatch's three-axis accelerometer at 10 Hz.



Figure 2. Label selection screen of smartwatch application.

3.2. Smartphone

The smartphone application collects data from the smartwatch and transmits them to the server, which shows results classified by gesture patterns based on the received data. The smartphone is connected to the smartwatch via Bluetooth and to the server via Wi-Fi. In this study, the Apple iPhone 6 is used because it offers good connectivity with Apple Watch.

3.3. Server

Compared with workstations, mobile computing platforms have limited processing power; hence, it is difficult to train a machine learning model using the processing power of a mobile device. To overcome these limitations, we have adopted a cloud system that uses a server for computationally-intensive tasks. The server plays two vital roles: training the classifier and classification. The classifiers that run on the server need to be trained by the collected dataset. The server uses the trained classifiers to recognize the gesture pattern using the transmitted data from the smartphone and sends the result, i.e., one of the 10 gestures predicted by the classification model, back to the smartwatch and smartphone. The server, running the CentOS, is equipped with Intel Xeon E5-2630 2.2 GHz CPU 2EA, 256 GB RAM, and GTX1080Ti GPU 4EA. The deep learning experiment in this study was implemented using Tensorflow [26] and Keras [27].

4. Gesture Pattern Recognition with CNN

4.1. Gesture Patterns

In this study, we select and classify 10 patterns that can be used for the user interface design, as shown in Figure 3. On the two-dimension space, various patterns involving 9 dots are considered to determine 10 patterns.



Figure 3. Gesture patterns.

The method used to collect gesture patterns is shown in Figure 4. The participants in the experiment wear a smartwatch on the wrist that they mainly use and perform gestures from the given set of 10 gesture patterns, as shown in the Figure 3. The nine reference points in Figure 4 are offered for the participants' convenience and the 10 gesture patterns in Figure 3 are shown to the participants who then perform each gesture with the posture, hand shape, and motion that they want to take. In this paper, the gesture patterns are performed on a 2-dimensional flat surface like a desk or a flat plate.



Figure 4. How to collect the gesture pattern dataset.

4.2. Observation on Gesture Traces

A gesture trace is described as a time series of the acceleration measurements. Let A_x , A_y , and A_z be vectors of x, y, and z acceleration components according to the smartwatch axes. Let $a_i^{t_j}$ be the value measured from the accelerometer on the *i*-axis, and in time, t_j , which is the *j*-th sampling time. Let n be the number of samples within the whole gesture. Therefore, we have

$$A_x = \left(a_x^{t_1}, \dots, a_x^{t_n}\right),$$
$$A_y = \left(a_y^{t_1}, \dots, a_y^{t_n}\right),$$
$$A_z = \left(a_z^{t_1}, \dots, a_z^{t_n}\right).$$

The raw acceleration of the gesture pattern information collected using the smartwatch consists of A_x , A_y , and A_z . Figure 5 shows the one of raw acceleration series of gesture Patterns 1 and 10. It can be seen that due to the different pattern scenarios based on the type of gesture, the amount of runtime and acceleration values vary. For example, there were two rapid fluctuations in Pattern 1, whereas there were more than five fluctuations in Pattern 10. The difference in execution time between two gestures is about 1.5 s.



Figure 5. The graph of gesture 1 and 10 traces.

4.3. CNN

In this section, the proposed classification model for the 10 gesture patterns is explained. The CNN algorithm is used for classification. To use the CNN algorithm, the input data size would need to be

fixed. The input data size is fixed to 10 s. The collected pattern data set is 3 to 6 s long. Sampled at 10 Hz, the data at each axis has data from 30 to 60, each of which was set to 10 s long, that is, 100 in size. Thus, the total input data size is $1 \times 100 \times 3$ where each of the *x*, *y* and *z*-axis has 100 data. Insufficient data are padded with the value at the end. For example, the trace of Figure 5b is 5 s long and then blank data from 5 to 10 s are set to the value at 5 s.

Figure 6 shows the procedure of the designed CNN model. The machine learning classification model generally consists of the input, hidden, and output layers. The designed CNN model can be divided into the input, convolution, pooling, fully-connected, and output layers. The input layer enters the data to be classified based on the input format. The hidden layer consists of the convolution, pooling, and the fully-connected layers. The convolution layer is connected to some area of the input image and calculates the dot product between this connected area and its own weighted value. The pooling layer performs downsampling per dimension and outputs the decreased volume. In the fully-connected layer, all nodes are interconnected, and the result of each node is calculated by the matrix multiplication of the weight and adding a bias to it. In the output layer, all classes are converted into a probability via the Softmax function and are classified by the highest probability.



Figure 6. The overall architecture of the convolution neural network (CNN) used in the proposed algorithm.

The detailed structure of the designed CNN model is shown in Table 1. Conv, Pool, and FC are the acronyms of convolution, pooling, and the fully-connected layers, respectively. It also shows the size of the patch and stride used by each layer, the output size, and the number of parameters. The convolution layer starts its feature maps from nine and increases it by two times each. The pooling layer reduces the data size by half, at which point it uses the Max pooling method, which reduces the size of data, by selecting the largest value within the size of the filter. The fully-connected layer occurs three times in total, and through the two dropouts, it reduces overfitting. The dropout rates are set to 0.5.

To learn the CNN model, the study used the Rectified Linear Unit (ReLU) [28] as an activation function, cross-entropy as a cost function, and Adam [29] as an optimizer. Training minimizes the value of loss function with respect to parameters such as weights and biases using the optimizer and the parameters are updated repeatedly. The Adam optimizer is used since it is known to achieve good results fast [29]. The learning rate is set to 0.0001.

Layers	Patch Size/Stride	Number of Parameters	Output Size		
Input	-	-	$1 \times 100 \times 3$		
Conv1	$1 \times 3 \times 3 \times 9 + 9/1$	90	1 imes 100 imes 9		
Pool1	$1 \times 2/2$	-	$1 \times 50 \times 9$		
Conv2	$1 \times 3 \times 9 \times 18 + 18/1$	504	$1 \times 50 \times 18$		
Pool2	$1 \times 2/2$	-	1 imes 25 imes 18		
Conv3	$1 \times 3 \times 18 \times 36 + 36/1$	1980	1 imes 25 imes 36		
Pool3	$1 \times 2/2$	-	1 imes 13 imes 36		
Conv4	$1 \times 3 \times 36 \times 72 + 72/1$	7848	$1 \times 13 \times 72$		
Pool4	$1 \times 2/2$	-	$1 \times 7 \times 72$		
Conv5	$1 \times 3 \times 72 \times 144 + 144/1$	31,248	$1 \times 7 \times 144$		
Pool5	$1 \times 2/2$	-	$1 \times 4 \times 144$		
Conv6	$1 \times 3 \times 144 \times 288 + 288/1$	124,704	1 imes 4 imes 288		
Pool6	$1 \times 2/2$	-	$1 \times 2 \times 288$		
FC1	$576 \times 576 + 576$	332,352	576		
Dropout	-	-	-		
FC2	$576 \times 144 + 144$	83,088	144		
Dropout	-	-	-		
FC3	$144 \times 10 + 10$	1450	10		
Softmax	-	-	10		

Table 1. The detailed structure of the CNN used in the proposed algorithm.

5. Performance Evaluation

In this chapter, the gesture pattern recognition system designed in the previous chapter will be evaluated. The dataset used in the gesture pattern recognition system and performance evaluation index is explained, and the performance of the proposed model will be compared and discussed with the previous models.

5.1. Experiment Dataset

For the performance evaluation of the model, a total of 5000 experiment datasets were collected by making the ten participants perform every gesture 50 times for three days. Based on this dataset, the proposed gesture pattern recognition system was trained and tested. The training and testing process used a 5-fold validation algorithm to improve the reliability of the performance evaluation result. The number of the dataset collected by each gesture pattern is shown in Table 2.

Gesture Pattern	Number of Datasets
P1	500
P2	500
P3	500
P4	500
P5	500
P6	500
P7	500
P8	500
Р9	500
P10	500
Total	5000

Table 2. The number of a dataset according to gesture pattern types.

5.2. Performance Measures

For the effective performance evaluation of the proposed system, we used the following four indicators: accuracy, precision, recall, and F1-score [30]. Equations (1)–(4) and Table 3 show how

accuracy, precision, recall, and F1-score are derived, respectively. These four expressions are the most frequently used performance indicators for machine learning models.

Accuracy =
$$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$
, (1)

Precision =
$$\frac{(TP)}{(TP + FP)}$$
, (2)

$$\operatorname{Recall} = \frac{(TP)}{(TP + FN)},$$
(3)

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}.$$
 (4)

Table 3. Confusion matrix of two-class classification.

		True Condition lation Condition Positive Condition Negative			
	Total Population				
Predicted Condition	Predicted condition positive	True Positive (TP)	False Positive (FP)		
Treatered Condition	Predicted condition negative	False Negative (FN)	True Negative (TN)		

5.3. Proposed Model Evaluation

The performance evaluation results of the proposed gesture pattern recognition system are shown in Table 4.

Table 4. The score of the proposed classification model.

Accuracy	Precision	Recall	F1-Score
97.3%	97.36%	97.32%	97.32%

The detailed results of the proposed classification model are as follows: all four performance indices are very high at about 97%. In Table 5, P1, P6, P9, and P10 show an extremely high accuracy rate of 99%, classifying almost all gestures. The classification performance of these four patterns is excellent. P7 and P8 show an estimation rate of 97%, and P3 shows a lower estimation rate of 92.2%. The detailed explanation of the confusion matrix shows that P2 and P3, and P5 and P8 have similar characteristics; therefore, we may get confused between them. Furthermore, P1 was confused with P2, P2 and P3 with P9, P4 with P3 or P9, P5 with P6 or P8, and P7 with P4, P9 or P10. Generally, all patterns show an equally high estimation rate.

Table 5. The confusion matrix of the proposed classification model.

Actual Class	Predicted Class									
	P1	P2	P3	P4	P5	P6	P 7	P8	P9	P10
P1	99.1%	0.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
P2	0.0%	96.3%	2.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.9%	0.0%
P3	0.0%	4.9%	92.2%	0.0%	0.0%	1.0%	0.0%	0.0%	1.9%	0.0%
P4	0.0%	0.0%	1.0%	98.0%	0.0%	0.0%	0.0%	0.0%	1.0%	0.0%
P5	0.0%	0.0%	0.0%	0.0%	96.7%	1.1%	0.0%	2.2%	0.0%	0.0%
P6	0.0%	0.0%	0.0%	0.0%	0.0%	99.0%	0.0%	0.0%	1.0%	0.0%
P7	0.0%	0.0%	0.0%	1.0%	0.0%	0.0%	97.1%	0.0%	1.0%	1.0%
P8	1.0%	0.0%	0.0%	0.0%	1.0%	1.0%	0.0%	96.9%	0.0%	0.0%
P9	0.0%	1.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	99.0%	0.0%
P10	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.1%	98.9%

5.4. Comparison with Other Classification Algorithms

In this section, comparison of the performance evaluation results between the proposed classification model and other classification algorithms, as discussed in Section 2, is made. In this performance evaluation, the classification model uses the SVM [17] and DTW [10] algorithms, which have been used in most of the previous studies.

One of the most important problems of the SVM algorithm is the identification of the features that will represent the data. In this study, we used 34 features which were selected from a previous paper [15]. They include time domain features (gesture time length, zero-crossing rate, mean, standard deviation, maximal, and minimal), frequency domain features (period and energy), and singular value decomposition (SVD) features (first and second column vector of the rotation matrix and relative values of the singular values). An optimal tuning of hyperparameters in SVM-based classification model is also key to achieving high classification accuracies. To obtain the lowest error rate, we performed a grid-search over the regularization parameter (C) between 1 and 10,000 and the gamma parameter (γ) between 0.0001 and 1. The radial basis function (RBF) has been used as the kernel of the SVM-based classification model.

The DTW method measures the distance between the test gesture patterns dataset and the template gesture patterns dataset of each class and selects the class with the minimum distance. In this study, the DTW-based classification model uses the nearest neighbor algorithm after obtaining DTW values between test data and all data in the training dataset.

The graphs representing the performance of each classification model is shown in Figure 7. Overall, the performance of the CNN-based classification model we proposed in this article is superior in comparison with other classification models, and the SVM-based classification model shows the poorest performance. On the contrary, in comparison with the SVM-based classification model, the CNN-based classification model shows an increase in accuracy of 8.1%, precision of 8.0%, recall of 8.1%, and F1-score of 8.1%; in comparison with the DTW-based classification model, it also shows an increase in accuracy of 1.3%, precision of 1.3%, recall of 1.3%, and F1-score of 1.3% [15]. In comparison with the other classification models, the CNN-based classification model shows superior performance in all performance indices. The difference in accuracy, precision, recall, and F1-score between the SVM-based and DTW-based classification models is 6.8%, 6.7%, 6.8%, and 6.8%, respectively.



Figure 7. The results of the various algorithms.

5.5. Discussion

The classification model proposed in this paper classifies 10 gesture patterns at an average accuracy of 97.3%. Before acquiring the study results, it is assumed that the classification would

cause some confusion since the classification model would detect and classify the characteristics of patterns through the accelerometer on a smartwatch. However, our results refute this assumption. For example, it is assumed that P1 and P2 would be confused as their gesture patterns are simple and the movement mechanisms are similar to each other; however, the classification model did not confuse the two patterns. Similarly, it is assumed that P10 would not be confused at all owing to its complex pattern. However, the results show that it is confused with P9 instead. While the results are different from the estimation, it was verified that the proposed classification model classified each pattern at a very high estimation rate. Furthermore, in comparison with the classification models used in the existing studies, the proposed model shows superior performance. Of course, it is noteworthy that the proposed algorithm requires more sample data to achieve good performance compared to the SVM and DTW algorithms, since it is based on a deep neural network model.

6. Conclusions

Owing to the recent advancements in the field of wearable device technology and an increase in its usage, there is a growing need for comfortable user interface design. However, due to various limitations of wearable devices, such as the size and volume, it is difficult to install additional input devices or sensors. Accordingly, many studies have been conducted to realize a precise and detailed user interface using various sensors that are already installed in wearable devices to track and classify user movements. Existing studies have classified simple patterns using the DTW algorithm or relatively light machine learning algorithms. However, to prove a more diverse range of functions and improve the user interface, it is necessary to increase the number of types of gesture patterns that can be recognized and improve the accuracy of gesture pattern recognition.

In this paper, the proposed classification model was used to classify 10 gesture patterns with an accuracy of 97.3%, and due to the nature of the deep-learning technology, the more data able to be accumulated, the better the accuracy will be. Not only will the advancement of such technology help realize various user interface designs, but it will also make it possible to realize user interfaces that reflect the users' individual personality. Furthermore, it will allow for the expansion of various other functionalities and utilities, such as signature, personal security, or user identification, among others, because it enables lightweight authentication based on the physical manipulation of the device. It is believed that more diverse services will be offered based on the proposed classification model [13,31,32]. To this end, future researches may focus on gesture pattern recognition with an increased number of reference points or three-dimensional gestures.

Author Contributions: M.-C.K. and S.C. conceived and designed the experiments; M.-C.K. and G.P. performed the experiments; M.-C.K., G.P., and S.C. analyzed the data; M.-C.K. and S.C. wrote the paper; and all authors proofread the paper.

Funding: This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) (No. 2016R1A5A1012966).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- 1. Oppitz, M.; Tomsu, P. Internet of Things. In *Inventing the Cloud Century*, 1st ed.; Springer: Cham, Switzerland, 2018; pp. 435–469.
- 2. Conti, M.; Dehghantanha, A.; Franke, K.; Watson, S. Internet of Things security and forensics: Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *78*, 544–546. [CrossRef]
- Data and Analysis. Business Consulting. Custom Research. Available online: http://www.strategyanalytics. com/ (accessed on 22 March 2018).
- Ko, D.; Kwon, H.; Park, J.; Han, J.; Kwon, J.; Kim, J.; Lee, H.; Kang, P. Smart Watch and Method for Controlling the Same. U.S. Patent Application No. 9,939,788B2, 10 April 2018.
- 5. Abramov, A. A Wearable Smart Watch with a Control-Ring and a User Feedback Mechanism. U.S. Patent Application No. 20,180,052,428A1, 22 February 2018.

- Ruan, S.; Wobbrock, J.O.; Liou, K.; Ng, A.; Landay, J.A. Comparing Speech and Keyboard Text Entry for short Messages in Two Languages on Touchscreen Phones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2017, 1, 159. [CrossRef]
- 7. Apple Inc. Siri. Available online: http://www.apple.com/ios/siri/ (accessed on 22 March 2018).
- 8. Samsung Electronics CO., LTD. Available online: http://www.samsung.com/global/galaxy/apps/bixby (accessed on 22 March 2018).
- Chung, J.; Oh, C.; Park, S.; Suh, B. PairRing: A Ring-Shaped Rotatable Smartwatch Controller. In Proceedings of the Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, Montréal, QC, Canada, 21–26 April 2018; Volume 10.
- 10. Müller, M. Dynamic time warping. In *Information Retrieval for Music and Motion*, 1st ed.; Springer: Berlin, Germany, 2007; pp. 69–84.
- 11. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn.* **2011**, *12*, 2825–2830.
- 12. Liu, J.; Zhong, L.; Wickramasuriya, J.; Vasudevan, V. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive Mob. Comput.* **2009**, *5*, 657–675. [CrossRef]
- Lee, W.H.; Liu, X.; Shen, Y.; Jin, H.; Lee, R.B. Secure Pick Up: Implicit Authentication When You Start Using the Smartphone. In Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies—SACMAT'17, Indianapolis, IN, USA, 21–23 June 2017; ACM Press: New York, NY, USA, 2017; pp. 67–78.
- 14. Wu, J.; Pan, G.; Zhang, D.; Qi, G.; Li, S. Gesture recognition with a 3-D accelerometer. In *Ubiquitous Intelligence and Computing*; Springer: Berlin, Germany, 2009; pp. 25–38.
- 15. Hong, F.; You, S.; Wei, M.; Zhang, Y.; Guo, Z. MGRA: Motion Gesture Recognition via Accelerometer. *Sensors* **2016**, *16*, 530. [CrossRef] [PubMed]
- Chen, Y.; Gao, W.; Ma, J. Hand Gesture Recognition Based on Decision Tree. In Proceedings of the Int. Symposium on Chinese Spoken Language Processing—ISCSLP 2000, Beijing, China, 13–15 October 2000; pp. 299–302.
- 17. Steinwart, I.; Christmann, A. Support Vector Machines; Springer Science & Business Media: Berlin, Germany, 2008.
- Xie, M.; Pan, D. Accelerometer Gesture Recognition. 2014. pp. 1–5. Available online: https://cs.stanford. edu/~eix/ (accessed on 22 March 2018).
- 19. Xu, H.; Zhou, Y.; Lyu, M.R. Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones. *Symp. Usable Priv. Security* **2017**, *14*, 187–198.
- 20. Sitova, Z.; Sedenka, J.; Yang, Q.; Peng, G.; Zhou, G.; Gasti, P.; Balagani, K.S. HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users. *IEEE Trans. Inf. Forensic Secur.* **2016**, *11*, 877–892. [CrossRef]
- 21. Lee, K.T.; Yoon, H.; Lee, Y. Implementation of smartwatch user interface using machine learning based motion recognition. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 807–809.
- 22. Wang, Z.; Yang, Z.; Dong, T. A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities and Monitor Vital Signs in Real Time. *Sensors* **2017**, *17*, 341.
- Son, D.; Lee, J.; Qiao, S.; Ghaffari, R.; Kim, J.; Lee, J.E.; Song, C.; Kim, S.J.; Lee, D.J.; Jun, S.W.; et al. Multifunctional wearable devices for diagnosis and therapy of movement disorders. *Nat. Nanotechnol.* 2014, 9, 397–404. [CrossRef] [PubMed]
- 24. Um, T.T.; Babakeshizadeh, V.; Kulic, D. Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2385–2390.
- 25. Kwon, M.C.; Choi, S. Recognition of Daily Human Activity Using an Artificial Neural Network and Smartwatch. *Wirel. Commun. Mob. Comput.* **2018**. [CrossRef]
- 26. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. Tensorflow: A system for large-scale machine learning. *OSDI* **2016**, *16*, 265–283.
- 27. Keras: The Python Deep Learning Library. Available online: https://keras.io/ (accessed on 25 July 2018).
- 28. Nair, V.; Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010.
- 29. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv, 2014; arXiv:1412.6980.

- 30. Olson, D.L.; Delen, D. Advanced Data Mining Techniques; Springer Science & Business Media: Berlin, Germany, 2008.
- 31. Buriro, A.; Crispo, B.; Delfrari, F.; Wrona, K. Hold and sign: A novel behavioral biometrics for smartphone user authentication. In Proceedings of the IEEE Security and Privacy Workshops (SPW), San Jose, CA, USA, 23–25 May 2016.
- 32. Griswold-Steiner, I.; Matovu, R.; Serwadda, A. Handwriting watcher: A mechanism for smartwatch-driven handwriting authentication. In Proceedings of the 2017 IEEE International Joint Conference on Biometrics (IJCB), Denver, CO, USA, 1–4 October 2017.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).