Human activity detection based on multiple smart phone sensors and machine learning algorithms

Conference Paper · May 2015

DOI: 10.1109/CSCWD.2015.7231023

Weiming Shen
National Research Council Canada

Xianbin Wang
The University of Western Ontario

Human Activity Detection Based on Multiple Smart Phone Sensors and Machine Learning Algorithms

Xizhe Yin, Weiming Shen, Jagath Samarabandu, Xianbin Wang
Dept. of Electrical & Computer Engineering
University of Western Ontario
London, Ontario, Canada
{xyin53; wshen; jagath; xianbin.wang}@uwo.ca

***Abstract*—This paper presents our recent work on human activity detection based on smart phone embedded sensors and learning algorithms. The proposed human activity detection system recognizes human activities including walking, running, and sitting. While walking and running can be recorded as daily fitness activities, falling will also be detected as anomalous situations and alerting messages can be sent as needed. Embedded sensors including a tri-axial accelerometer, tri-axial linear accelerometer, gyroscope sensor, and orientation sensors are used for motion data collection. A two-stage data analysis approach is used for prediction model generation: short period statistical analysis (max, min, mean, and standard deviation) and long period data analysis using machine learning. The system is implemented in an Android smart phone platform.**

Keywords—activity detection; sensors; machine learning; smart phones

I. Introduction

Human activity detection is regarded as one of the most important issues in smart home environments as it has a wide range of applications including healthcare, daily fitness recording, and anomalous situations alerting. Some simple human activities (walking, running, sitting, etc.) are essential for understanding complex human behaviors. Also, the detection of activities such as walking and sitting can provide useful information about a person's health condition.

Nowadays, cell phone is not only a tool for communication. Smart phone and smart phone-based applications are used everywhere. Most smart phones have a tri-axial accelerometer and gyroscope as well as orientation sensors. All these sensors can tell us the acceleration and rate of rotation on and around three physical axes (X, Y, and Z). With these embedded sensors, we can identify different kinds of activities instead of using other wearable sensors attached to human body.

In this paper, we present a smart phone-based human activity detection system. The data are collected by the embedded sensors. Some machine learning algorithms (J48, SVM, Naïve Bayes, and Multilayer Perceptron) are employed to make classifications. The system is implemented in an Android smart phone platform.

This paper is organized as follows. Section II provides a brief literature review. The proposed methods are introduced in Section III. Experiment results and analyses are given in Section IV. Conclusions are made in Section V.

II. Related Work

Human activity detection has been widely studied in the literature. Some methods and prototypes have been developed according to the different kinds of information and algorithms. The data used for detection include video data or sensor data (acceleration, orientation, and angular speed, etc.). The sensors could be on-board smart phone sensors or sensors installed in wearable devices. In addition, different methods and algorithms can be used for activity detection. In [1], probability-based algorithms to build activity models are studied. The hidden Markov model (HMM) and the conditional random field (CRF) are among the most popular modeling techniques.

Zhou et al. [2] proposed an indoor human activity monitoring system based on continuous video monitoring and intelligent video processing. They claimed that the reason for using camera data is that it can provide a rich and unique set of information that cannot be obtained from other types of sensors. However, camera-based methods require continuously monitoring a person's activities, which need a large number of storage space and computation resource. In addition, people may feel uncomfortable being watched by cameras continuously [3].

Dong et al. [4] presented a wearable sensor network that was designed for human activity detection. In their system, each of three sensors, which were worn on ankle, thigh, and wrist, was equipped with an ADXL202 accelerometer. The mean and entropy of acceleration were computed over time windows and were used for activity detection. Yin et al. [5] proposed another abnormal activity detection system based on wearable sensors. In their approach, a one-class support vector machine (SVM) and a kernel nonlinear regression (KNLR) were employed for the two-phase training and activity detection. Three MTS310CA sensor boards were attached to different parts of a human body. And seven features were selected as inputs (light, temperature, microphone, two-axis accelerometer, and two-axis accelerometer). Their work makes it possible for abnormal activity models to be automatically derived without the need to explicitly label the abnormal training data. Then, Curone et al. [6] used one wearable tri-axial accelerometer for activity detection. Signal magnitude area (SMA) [7] was introduced in their method, which was an effective measurement for identifying activity intensity. Al-Ani et al. [8] combined wavelet and hidden Markov models for on-line human activity detection. Their experiments were also conducted by wearable sensors attached to a person. Chen and Huang [9] described a method of detecting human activities through classifying multi-modal sensors data. Seven features (MagX, MagY, Micphone, Light, AccelX, AccelY, and Temperature) were selected for classification. Then they tested some popular algorithms (Naive Bayes, J48, Random Tree, K Star), which showed satisfying results. Since body sensor networks (BSNs) based activity detection systems usually have difficulties in real-world applications due to the programming complexity and the lack of high-level software abstractions, Fortino et al. [10] developed a programming framework called signal processing in node environment (SPINE) to support rapid development of BSN applications.

As mentioned above, with ever increasing computing power, convenient Internet connections, and numerous mobile applications, smart phones have been an indispensable role in our daily life. What's more, even low-end smart phones have a set of sensors (accelerometer, GPS, and gyroscope, etc.), which make it possible to use a phone to detect human activities. However, the sensing task performed by the phone-based sensors is different from that by other wearable sensor devices. The wearable devices are fixed on human body, while smart phones are usually placed in a person's pocket loosely and have different orientations. In [11], a mobile phone-based fall detection system was proposed and implemented in Android platform. In their experiments, the mobiles phones were still attached to different locations of human body. In [12], a hierarchical SVM classifier was used for recognizing walking, going-upstairs, going-downstairs, running, and motionless. The features selected from accelerometer data were standard deviation of Y axis, correlation of Y, Z axis, autoregressive fitting of Y axis, and the SMA. Also, the mean, standard deviation, and skewness of pitch were selected for classification. In [13], a gait-based-recognition was proposed to recognize walking activity. The wavelet transform was employed to extract features from raw data and the k Nearest Neighbors (kNN) algorithm was used to perform the classification.

III. Human Activity Detection Methods

A. Smart Phone Based Approach

This paper proposes to use smart phones to detect human activities including walking, running, and sitting. As "standing still" or "putting the phone on a table" is similar to the situation of "sitting", we use "sitting" in this paper to represent the motionless state. Also, we assume that people carry a smart phone in pockets. The smart phone based application performs the tasks of collecting sensor data and making predictions.

B. Data Collection

In this study, the signals detected from the embedded accelerometer, linear accelerometer, gyroscope, and orientation sensors in smart phones are used to infer different human activities. To collect these sensor data, one subject puts a smart phone in his/her pocket and performs some activities. Preliminary data collection and analyses show no difference when the smart phone is placed in different orientations in one pocket or in different pockets. On the other hand, this would make the smart phone based detection system simpler and more practical. Therefore, the subject does not need to care about how to place the smart phone in his/her pockets during the data collection. An android application is running during the experiments. Sensor data are saved as CSV files. In order to increase the classification accuracy, data recorded in the beginning and end of each activities are trimmed from the data files. Due to the Android APIs [14], we record the sensor data when the sensor values are changed every 20 times.

C. Feature Extraction

We monitor the changes of accelerometer, orientation, gyroscope, and linear acceleration values. And every 20 times the sensors' values are changed, the 20th sensors' values are recorded. The sampling frequency is about 20Hz. In addition, four time domain features (maximum, minimum, mean, and standard deviation of every 20 values) are computed and stored. So a fixed window length without overlapping is used in the feature extraction stage. For each sensor, we regard its three axes as three individual features. As we monitor four sensors and there are five time domain features for each axis as well as three axes for each sensor, a total of 60 features are extracted for each instance (as shown in Fig. 1). The activity types are labeled manually in each file.

Features from accelerometer, gyroscope, and linear accelerometer sensors (every 20 values)
○ The 20th sensor data of X, Y, and Z-axis
○ Maximum of X, Y, and Z-axis
○ Minimum of X, Y, and Z-axis
○ Mean of X, Y, and Z-axis
○ Standard deviation of X, Y, and Z-axis
Features from orientation sensor (every 20 values)
○ The 20th sensor data of azimuth, pitch, and roll
○ Maximum of azimuth, pitch, and roll
○ Minimum of azimuth, pitch, and roll
○ Standard deviation of azimuth, pitch, and roll

Fig. 1. Feature extraction.

D. Classifiers

Four supervised machine learning algorithms are described below. Performances of these classifiers will be discussed in Section IV.

1) J48

The output of J48 algorithm is a decision tree. Let S be our data set for training. $freq(C_i, S)$ is the number of instances in S that belongs to class C_i. The number of instances in data set S is $|S|$. The entropy of set S is

$$Info(S) = -\sum_{i=1}^{k}\left(\left(\frac{freq(C_i,S)}{|S|}\right)\cdot \log_2\left(\frac{freq(C_i,S)}{|S|}\right)\right)$$

If set S is partitioned in accordance with n outcomes of one attribute X, then

$$Info_X(S) = \sum_{i=1}^{n}\left(\left(\frac{|S_i|}{|S|}\right)\cdot Info(S_i)\right)$$

The information gain is

$$Gain(X) = Info(S) - Info_X(S)$$

The attribute with the highest normalized information gain is chosen to make the decision.

After a decision tree model is generated, the path to each leaf can be transformed into IF-THEN rules, which is convenient for future applications.

2) SVM

As a supervised machine learning algorithm, the support vector machine (SVM) can construct a hyperplane and separate two classes, which tries to maximize the margin between the nearest data points in each class. The hyperplane can be stated as $w^T x + w_0 = 0$. The two classes are labeled as -1 and 1. Let $\{x_i\}$ be the training set, and $\{y_i\}$ represents the output of SVM for a new point x_i. So $y_i \in \{-1,1\}$. If $w^T x_o + w_0 > 0$, then $y_i = 1$. If $w^T x_o + w_0 < 0$, then $y_i = -1$. To determine the hyperplane, we have to solve the equation that maximizes the distance between the closest points of each class: $\arg\max\min\{\|x - x_i\|: x \in R^d, w^T x + w_0 = 0\}$, where the variables are w and w_0, i is from 1 to N.

Our problem has more than two classes. So before applying SVM, we have to extend the standard SVM to multiclass SVM. There are two strategies, one-against-one and one-against-all strategy, to deal with multiclass situation.

3) Naïve Bayes

Naïve Bayes classifier is based on the Bayesian theorem, which says $p(c_j|d) = p(d|c_j)p(c_j)/p(d)$. $p(c_j|d)$ is the probability of instance d being in class c_j. $p(d|c_j)$ is the probability of instance d given in class c_j. $p(c_j)$ is the probability of occurrence of class c_j. $p(d)$ is the probability of instance d occurring. The advantages of Naïve Bayes classifier are that it is fast to train and classify. However, it assumes independence of features.

4) Multilayer Perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network, which usually contains three or more layers. For node i, which is connected to node j in its following layer, has a weight w_{ij}. A MLP is trained through backpropagation. And the weights are updated according to the amount of error between the output and the desired output.

IV. Experimental Results and Analyses

In this section, we test 4 different classifiers (J48, Naïve Bayes, SVM, MLP) to recognize activities. The performances and some observations are discussed below.

A. Experimental Setup

We implement the activity detection system in an Android platform (Fig. 2). A smart phone is loosely placed in a person's pocket with different orientations. Using this application, we collect the tri-axial acceleration, linear acceleration, gyroscope data (rate of rotation in 3 axes), and orientation (azimuth, pitch, and roll) measurements for different activities, including walking, running, and sitting. Each data instance contains 60 features for classification. The training set is described in Table I. A 10-fold cross-validation method is employed for testing and WEKA framework is used in our study.

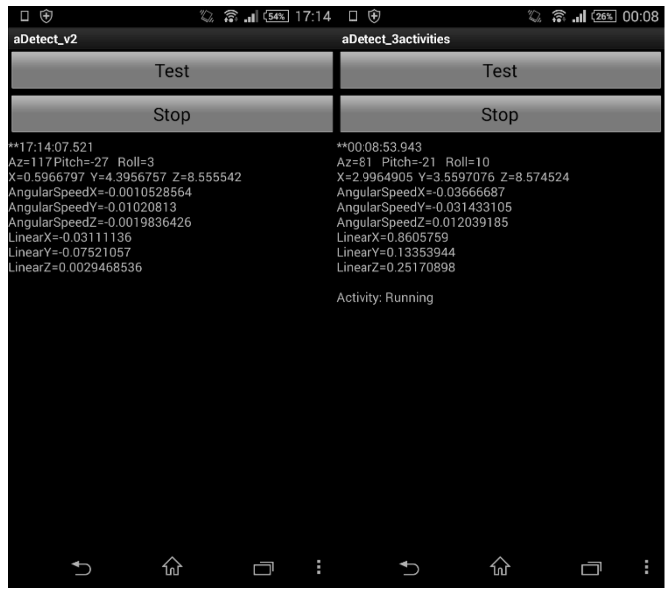

Fig. 2. The proposed activity detection system

TABLE I. Training Set

Activity Type	***Number of Instances***
Walking	486
Running	967
Sitting	462
Total	1915

B. Results and Analysis

The results in Table II show that all classifiers achieve an recognition accuracy of above 99% for the data set when identifying walking, running, and sitting. However, the multilayer perceptron classifier takes the longest time to build a model, which is a disadvantage if we want to perform the training step in a smart phone. For the SVM classifier, it is important to choose the appropriate parameters. If the radial basis function is used as the kernel type in SVM, the accuracy is only 73.94%. The optimal SVM parameters for one problem may be different from another.

J48 and Naïve Bayes are both simple and efficient classifiers. In addition, J48 can generate a decision tree for identifying activities (Fig. 3). From the decision tree, we observe that not all 60 features are necessary for identifying activities. Fig. 3 shows a generated J48 pruned tree. In this case, only 5 features (StdDev of linear acceleration in Z-axis, StdDev of orientation in Roll, StdDev of acceleration in Y-axis, StdDev of acceleration in X-axis, and StdDev of rate of rotation in Y-axis) are enough for detecting the three activities. This indicates that the standard deviation of a period of sensor data is more useful than the real sensor readings when the smart phone is placed in pockets casually. Standard deviations can tell us the intensity of different activities. For example, people may have different orientation when walking, running, and sitting. It is hard to identify these activities according to the orientation data. However, different activities can have different standard deviations of orientations in a short period of time, i.e., standard deviation is more sensitive than other values in identifying activities. The visualized data for 3 activities are shown in Fig. 4 to Fig. 9, which demonstrate the standard deviations in a period of time are the best features to identify activities.

```
J48 pruned tree
------------------

mLinearAccZ_StdDev <= 0.931683
|   Roll_StdDev <= 0.632456: Sitting  (448.0)
|   Roll_StdDev > 0.632456
|   |   Roll_StdDev <= 4.756837: Sitting  (14.0)
|   |   Roll_StdDev > 4.756837: Walking (2.0)
mLinearAccZ_StdDev > 0.931683
|   mSensorY_StdDev <= 2.268322
|   |   mGyroscopeY_StdDev <= 2.068244: Walking (469.0)
|   |   mGyroscopeY_StdDev > 2.068244
|   |   |   mSensorX_StdDev <= 4.175687: Walking (13.0)
|   |   |   mSensorX_StdDev > 4.175687: Running (4.0)
|   mSensorY_StdDev > 2.268322: Running (965.0/2.0)

Number of Leaves  :     7

Size of the tree :      13
```

Fig. 3. J48 decision tree

TABLE II. CLASSIFICATION RESULTS USING 60 FEATURES

Performance	Classifier			
	J48	*SVM*	*Naïve Bayes*	*Multilayer Perceptron*
Accuracy	99.0604%	99.4526%	99.1645%	99.8956%
Kappa statistic	0.9849	0.9908	0.9866	0.9983
Mean absolute error	0.0068	0.0038	0.0056	0.0019
Root mean squared error	0.0791	0.0619	0.0742	0.0274
Relative absolute error	1.647%	0.9228%	1.3422%	0.4518%
Root relative squared error	17.3685%	13.5859%	16.2899%	6.0067%
Time taken to build model (Seconds)	0.23	0.4	0.06	32.99

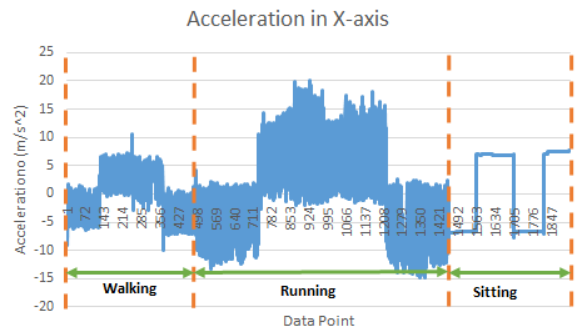

Fig. 4. Acceleration in X-axis

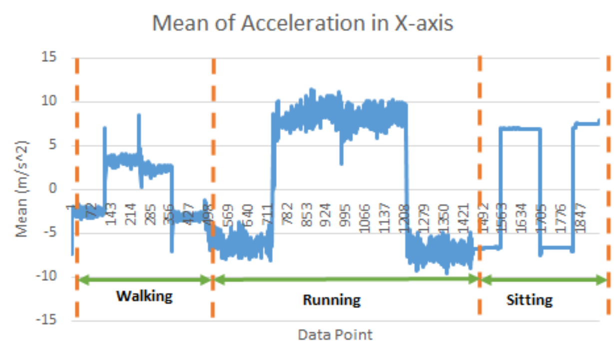

Fig. 5. Mean of acceleration in X-axis

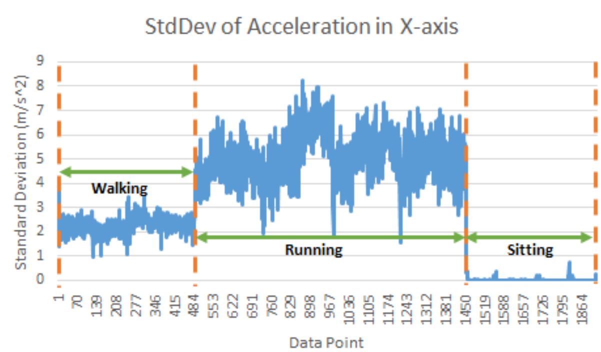

Fig. 6. Standard deviation of acceleration in X-axis

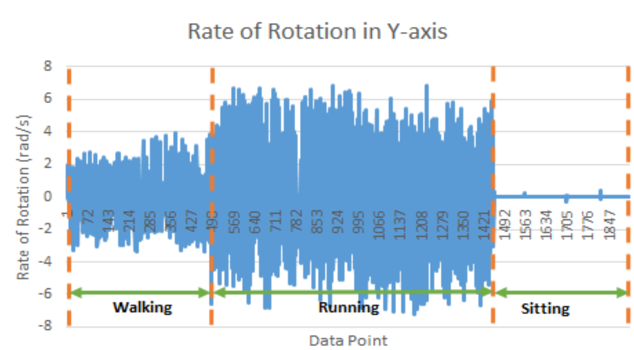

Fig. 7. Rate of rotation in Y-axis

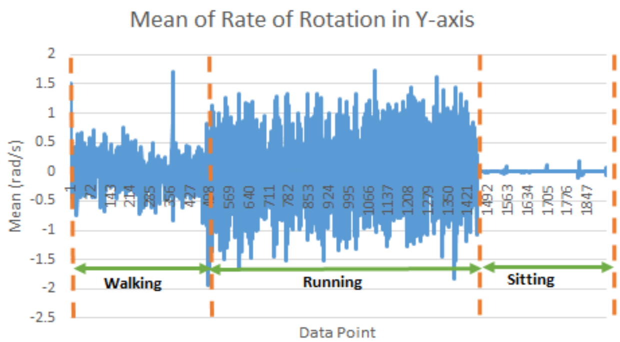

Fig. 8. Mean of rate of rotation in Y-axis

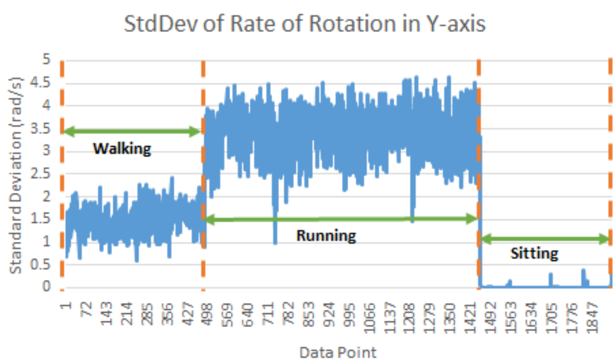

Fig. 9. Standard deviation of rate of rotation in Y-axis

As the standard deviations are more informative than the real sensor readings and other statistical values, we reduce the number of input features and use only 12 features for identifying activities. The selected features are shown in Table. III. Every 20 sensor readings are used for computing the standard deviation and the time interval of each data point is about 1 second. The classification results (as shown in Table. IV) are similar to those using 60 features. But the computational complexity and time consumption are reduced significantly.

TABLE III. REDUCED FEATURES

	12 Selected Features
o	Standard deviation of azimuth, pitch, and roll
o	Standard deviation of acceleration in X, Y, and Z-axis
o	Standard deviation of linear acceleration in X, Y, and Z-axis
o	Standard deviation of rate of rotation in X, Y, and Z-axis

TABLE IV. CLASSIFICATION RESULTS USING 12 FEATURES

Performance	Classifier			
	J48	***SVM***	***Naïve Bayes***	***Multilayer Perceptron***
Accuracy	99.6345%	99.5822%	99.3734%	99.7911%
Kappa statistic	0.9941	0.9933	0.9899	0.9966
Mean absolute error	0.0027	0.0028	0.0043	0.0027
Root mean squared error	0.0493	0.0528	0.0648	0.0279
Relative absolute error	0.6422%	0.6711%	1.0328%	0.6621%
Root relative squared error	10.8283%	11.5861%	14.2272%	6.1327%
Time taken to build model (s)	0.02	0.08	0.01	2.65

C. Extension

We added two more activities: going upstairs and going downstairs. Using the 12 features selected in Table III, we tested the performance of different machine learning algorithms for identifying the 5 activities. Results are shown in Table V.

Although they all have accuracies over 96%, SVM classifier outperforms J48 in identifying going upstairs and downstairs. The accuracies for J48 to identify going upstairs and downstairs are only 81.3% and 74.3%, while SVM reaches 97.3% and 85.1%. We believe that the performance of SVM can be improved if its parameters are tuned.

Even if J48 is a popular and efficient classifier, it has difficulties to model a large number of complex activities. What's more, it is easy to confuse walking, going upstairs and downstairs. As can be seen in Fig. 10, there is a large probability that going downstairs is misclassified as walking or going upstairs.

Once the model has been trained, it can be used to detect different activities. However, the input of the detection model is only one data instance, and it is defective. In our experiments, though the accuracy for identifying 3 activities reaches as high as 99%, the output can change quickly from one activity to another. To improve the robustness when using the trained model for real-time detection, one promising way is to consider the context information to determining the activity undertaken by a person.

TABLE V. CLASSIFICATION RESULTS USING 12 FEATURES (5 ACTIVITIES)

Performance	Classifier			
	J48	***SVM***	***Naïve Bayes***	***Multilayer Perceptron***
Accuracy (Overall Average)	96.8%	98.7%	97.8%	98.5%
Accuracy (going upstairs)	81.3%	97.3%	92.0%	95.5%
Accuracy (going downstairs)	74.3%	85.1%	87.1%	87.1%
Time taken to build model (s)	0.05	0.12	0.02	3.05

```
=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 472   4   0   4   6 |   a = Walking
   0 961   0   5   1 |   b = Running
   0   0 460   0   2 |   c = Sitting
   2   6   1  91  12 |   d = Upstairs
   8   0   2  16  75 |   e = Downstairs
```

Fig. 10. Confusion matrix for J48 classifier

V. CONCLUSION

Since smart phones have been indispensable in our daily life, smart phone-based human activity detection systems show the advantages compared with other wearable devices. Smart phones are easy to carry and have convenient access to the Internet. All these features make it possible for phone-based activity detection systems to record our daily activities and be our fitness trainer as well as send out alerting messages when anomalous activities are detected.

This paper proposed a novel two-stage data analysis approach for prediction model generation: short period statistical analysis (max, min, mean, and standard deviation) and long period data analysis using machine learning, which is believed to increase the accuracy of smart phone based human activity detection models. Use of standard deviation of sensor data in the final model makes it simpler and more practical.

Experimental results reveal that the proposed system can detect three simple activities (walking, running, and sitting) with a high accuracy (over 99%). Four popular machine learning algorithms (J48, SVM, Naïve Bayes, and MLP) are tested. Among them, J48 is a better solution. It is not only because a decision tree model is easy to compute, but also because it can be transformed into a set of IF-THEN rules, which is easy to be applied in smart phone applications with less computation. In addition, short period statistical values, especially the standard deviations, are found to be informative for detecting activities. Then we extend the system to identify two more activities, going upstairs and downstairs. The standard deviations of different sensors are still effective, and Naïve Bayes, MLP, and SVM classifiers can get satisfied classification results.

Our future work includes detecting additional human activities (e.g., bicycling, driving, or sitting in a car or other transportation vehicles), testing different ways of carrying smart phones, implementing self-learning algorithms, and developing advanced fall detection and fitness apps.

REFERENCES

[1] E. Kim, S. Helal, D. Cook, "Human Activity Recognition and Pattern Discovery," IEEE Pervasive Computing, 9(1), 48-53, 2010.

[2] Z. Zhou, X. Chen, Y. Chung, Z. He, T.X. Han, J.M. Keller, "Activity Analysis, Summarization, and Visualization for Indoor Human Activity Monitoring," IEEE Transactions on Circuits and Systems for Video Technology, 18(11), 1489-1498, 2008.

[3] M. Zhang, A.A. Sawchuk, "USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors," ACM Conference on Ubiquitous Computing (UbiComp'12), pp. 1036,1043, 2012.

[4] B. Dong, S. Biswas, "Wearable networked sensing for human mobility and activity analytics: A systems study," Fourth International Conference on Communication Systems and Networks (COMSNETS), pp. 1,6, 2012.

[5] J. Yin, Q. Yang, J.J. Pan, "Sensor-Based Abnormal Human-Activity Detection," IEEE Transactions on Knowledge and Data Engineering, 20(8), 1082-1090, 2008.

[6] D. Curone, G.M. Bertolotti, A. Cristiani, E.L. Secco, G. Magenes, "A Real-Time and Self-Calibrating Algorithm Based on Triaxial Accelerometer Signals for the Detection of Human Posture and Activity," IEEE Transactions on Information Technology in Biomedicine, 14(4), 1098-1105, 2010.

[7] C.V.C. Bouten, K.T.M. Koekkoek, M. Verduin, R. Kodde, J.D. Janssen, "A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity," IEEE Transactions on Biomedical Engineering, 44(3), 136-147, 1997.

[8] T. Al-Ani, Quynh Trang Le Ba, E. Monacelli, "On-line Automatic Detection of Human Activity in Home Using Wavelet and Hidden Markov Models Scilab Toolkits," IEEE International Conference on Control Applications (CCA'07), pp. 485,490, 2007.

[9] S. Chen, Y. Huang, "Recognizing human activities from multi-modal sensors," IEEE International Conference on Intelligence and Security Informatics (ISI'09), pp. 220,222, 2009.

[10] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, "Enabling Effective Programming and Flexible Management of Efficient Body Sensor Network Applications," IEEE Transactions on Human-Machine Systems 43(1), 115-133, 2013.

[11] S. Fang, Y. Liang, K. Chiu, "Developing a mobile phone-based fall detection system on Android platform," Computing, Communications and Applications Conference (ComComAp), pp. 143,146, 2012.

[12] J. Cho, J.T. Kim, T. Kim, "Smart Phone-based Human Activity Classification and Energy Expenditure Generation in Building Environments," 7th International Symposium on Sustainable Healthy Buildings (SHB'2012), 2012.

[13] M. Boyle, A. Klaustner, D. Starboinski, A. Trachtenberg, and H. Wu, "Poster: gait-based smartphone user identification," the 9th International Conference on Mobile Systems, Applications, and Services (MOBISYS'11), pp. 395,396, 2011.

[14] http://developer.android.com/reference/android/hardware/SensorEventListener.html