Human Activity Recognition using Android Smartphone

Usharani J, Dr. Usha Sakthivel
PG Scholar, Professor and HOD, Department of Computer Science,RRCE, Bengaluru-74
usharanismg@gmail.com, sakthivelusha@gmail.com,

ABSTRACT- Activity recognition is one of the most important technology behind many applications such as medical research, human survey system and it is an active research topic in health care and smart homes. Smart phones are equipped with various built-in sensing platforms like accelerometer, gyroscope, GPS, compass sensor and barometer, we can design a system to capture the state of the user. Activity recognition system takes the raw sensor reading from mobile sensors as inputs and estimates a human motion activity using data mining and machine learning techniques. In this paper, we analyze the performance of two classification algorithms i.e. KNN and Clustered KNN in an online activity recognition system working on Android platforms and this system will supports on-line training and classification using the accelerometer data only. Usually first we use the KNN classification algorithm and next we utilize an improvement of Minimum Distance and K-Nearest Neighbor classification algorithms, i.e. Clustered KNN . For the purpose of activity recognition, clustered KNN will eliminates the computational complexities of KNN by creating clusters (creating smaller training sets for each actions and classification will be performed based on these reduced training sets). We can predict the performance of these classifiers from a series of observations on human activities like walking, running, lying down, sitting and standing in an online activity recognition system. In this paper, we are intended to analyze the performance of classifiers with limited training data and limited accessible memory on the phones compared to off-line.

Keywords - Accelerometer; Activity Recognition ; Clustered KNN; KNN; Smartphone.

I. INTRODUCTION

Nowadays smartphones became more and more popular in human daily life. Most of the people used it for searching news, watching videos, playing games and accessing social network but there were many useful studies on smartphones. Activity recognition is one of the most important technologies behind many applications on smartphone such as health monitoring, fall detection, context-aware mobile applications, human survey system and home automation etc., Smartphone-based activity recognition system is an active area of research because they can lead to new types of mobile applications.

Understanding human activities creating a demand in health-care domain, especially in rehabilitation assistance, physiotherapist assistance, and elder care support services and cognitive impairment. Sensors will record and monitor the patient‘s activities and report automatically when any abnormality is detected, so, huge amount of resources can be saved. Other applications like human survey system and location indicator are all getting benefits from this study.

Training process is always necessary when a new activity is added in to the system. The same algorithm parameters are needed to be trained and fine-tuned when the algorithm runs on different devices with various built-in sensors. However, labeling a training data (time-series data) is a time consuming procedure and it is not always possible to label all the training data by the users. As a result, we present an active learning technique to accelerate the training process. Given a KNN classifier, an active learning technique intuitively queries the unlabeled training samples and learns the parameters from the correct labels answered by the human. In this way, users will label only the samples that the algorithm demanded to do and the total amount of required training samples is reduced.

HAR system takes the raw sensor readings from mobile sensors as inputs and predicts human motion activity, this can be done by leveraging smartphone with various sensors, including accelerometers, compass sensor, GPS, light sensors, gyroscope, barometer etc., Due to its unassertive, none/low installation cost and easy-to-use, smart phones are becoming the main platform for human activity recognition**.** In this paper, we focus on robust human activity recognition using 3-dimentional accelerometer and gyroscope on smart phones.

In this paper, we are also interested in analyzing the performance of classifiers with limited training data considering the limited memory available on the phones. In this system, we can collect the training data in a few minutes and it can be directly used for classification steps, which reduce the burden on the users. Being one of the first Android applications used for human activity recognition is another important motivation for this study. In the literature, it has been reported that minimum distance classifier does not work well when used alone. KNN results are always better than minimum distance classifier in terms of accuracy. However, KNN requires high computational burden so, it is not an online classifier and due to limited resources on smart phone, it does not appear as a preferable method.

The rest of the paper is organized as follows. We describe Related work in Section II and Section III will describes the Android Smartphone sensors used in activity recognition system. Section IV describes Core Techniques and the human activity recognition on Android Smartphone is described in Section V. Result and Discussion is described in Section VI. Finally, we conclude the paper in section VII.

II. RELATED WORK

Human activity recognition on smartphone is an active research area. Most of related works focus on analyzing the performance of classification algorithms such as: Decision Trees, Naïve Bayes, Nearest Neighbor algorithms, Support Vector Machines, Hidden Markov Chain, Multi-Layer Perceptron and Random Forrest. There are not much work investigated on features selection of dataset. Jun Yang and et al. extracted orientation-independent features from three feature sets, including horizontal, vertical and magnitude features. Each feature set consists of mean, standard deviation, zero cross rate, 75 percentile, interquartile, spectrum centroid, entropy. The authors used Attributed Selection filters to give 7 feature subsets and evaluate recognition accuracy on these subsets. As a result, the accuracy of classifiers on each subsets are lower than with all features, i.e. Decision Tree equals to 90.4% (all features: 90.6%), Naïve Bayes equals to 68.3% (all features: 68.7%). Sian Lun Lau and et al. used common four features mean, standard deviation, energy of the Fast Fourier Transform and correlation. They combined features to 3 groups: group G1 includes average and standard deviation of values of each axis and all three axes, group G2 includes the average and the standard deviation of FFT coefficients of each axis and all three axes, group G3 includes all four features of each axis and all three axes. However, they used simple features and combined features into groups manually. Ville Kononen and et al. used two feature selection methods, including Sequential Forward Selection and Selection to select features from accelerometer and heart rate signals and evaluate complex classification compared with simple classification. However, they used the feature selection method to select features and compared accuracy of classifier on that features and recognition accuracy of classifier range from about 60% to 90%.

Different from other work, in this paper we do not only remove the irrelevant and redundant features but also remove redundant instances. As a result, the system will achieve better recognition whereas the training dataset is reduced significantly.

III. ANDROID SMARTPHONE SENSORS

Android devices have built-in sensors that measures motion (accelerometers, gravity sensors, and gyroscope), orientation (magnetometers.), and various environmental conditions (barometers, photometers, and thermometers). These sensors are capable of providing raw data with high accuracy, and are useful if we wanted to monitor three-dimensional device movement or positioning, or we wanted to monitor changes in the ambient environment near a device.

The Android sensor framework allows us to access many types of sensors, such as hardware-based and software-based sensors. Hardware-based sensors are physical components built into a smartphone and they derive their raw data by directly measuring specific environmental parameters, such as acceleration, geomagnetic field strength, or angular change. Whereas, software-based sensors (linear acceleration sensor and the gravity sensor) are not physical devices, although they imitate hardware-based sensors. Software-based sensors derive their raw data from one or more hardware-based sensors and are sometimes called virtual sensors or synthetic sensors.

The android sensor framework uses a standard 3-axis coordinate system to express data values. For most sensors, the coordinate system is defined relative to the device's screen when the device is held in its default orientation (see fig 1). When a device is held in its default orientation, the X axis is horizontal and points to the right, the Y axis is vertical and points up, and the Z axis points toward the outside of the screen face. In this system, coordinates behind the screen have negative Z values.

1 Accelerometer

Accelerometer is a latest technology which has upgraded the user experience in smartphones and it measures the acceleration force applied to a device on all three physical axis (x, y, and z), including the force of gravity. It changes the orientation and adjusts the screen to proper viewing, when user changes the orientation from landscape/horizontal to portrait/vertical and vice-versa. The smartphone physical position can be determined by 3-way axis device. Fig 1 shows the Accelerometer axes on smartphone. The raw data from the accelerometer is represented in a set of vectors: Acc i=<x i, y i, z i> , where i=(1,2,3,….). A time stamp can also be returned with these 3-axis readings.

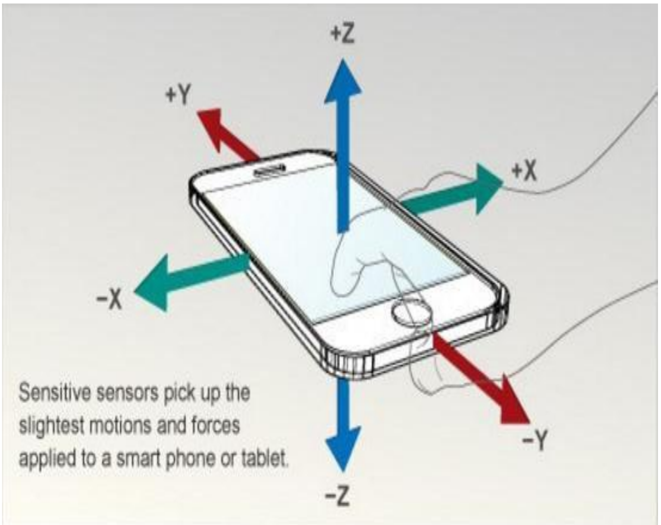

Fig 1: Accelerometer axes on smartphone.

2 Compass sensor

The digital Compass is a traditional tool to detect the direction with respect to the north-south pole of the earth's magnetic field. Compass functionality in smartphones is usually based on more sophisticated sensor called a magnetometer; it is used to measure the strength and direction of magnetic fields. Fig 2 shows the compass reading display screen on a Smartphone. By analyzing Earth's magnetic field, the sensor allows a phone to determine its orientation with high accuracy. The raw data reading from a compass sensor is the float number between 0º and 360º. It begins from 0º as the absolute north and the actual reading indicates the angle between present smart phone direction and the absolute north in clockwise. The Pro version of Smart Compass adds a speedometer and the option to send GPS coordinates via SMS or email. Compass reading can be used to detect the direction change in the human motion such as walking.

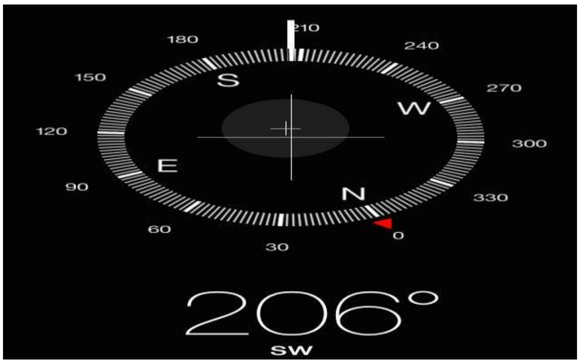

Fig 2 : Compass Sensor on Smartphone‘s.

3 Gyroscope

The gyroscope is a device, which adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist and it is primarily used for navigation and measurement of the angular rotational velocity. And it also uses earth‘s gravity to help determine orientation. Gyroscope measures the phone‘s rotation rate by detecting the roll, pitch, and raw motions of the smart phones along the x, y, and z axis, respectively. The axes directions are shown in Fig. 3. The raw data from a gyroscope is the rate of the rotation in rad/s (radian per second) around each of the three physical axes: Rotation i =<xi ; y i ; z i >; I =(1,2,3,..........) . In activity recognition search, gyroscope is used to assist the mobile orientation detection.

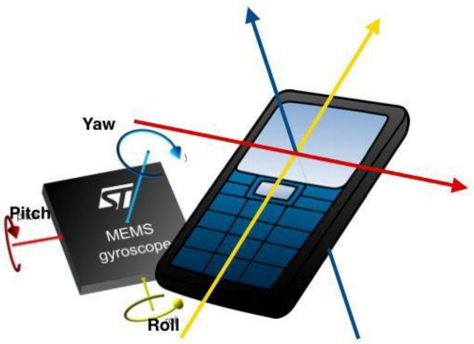

Fig 3: Gyroscope Three Axes on Smartphone‘s.

4 Barometer

Barometer is a device equipped on most of the advanced smart phones. It measures the atmospheric pressure of the environment where the sensor is placed in. So, barometer reading can be used to indicate the user‘s position change in localization related activity recognition. Barometers senses air pressure and are used in smartphones to determine relative elevation - measuring stairs climbed and so on. It should be able to take air pressure measurements and then help forecasters analyze where troughs, high-pressure zones, and frontal boundaries are.

IV. CORE TECHNIQUES

1 KNN

K-Nearest Neighbor is a supervised learning algorithm where the result of new instance query is classified based on majority of K-Nearest Neighbor category and it is one of the most popular algorithms for activity recognition. The purpose of KNN algorithm is to stores all available objects and classifies a new object based on attributes and training samples. The KNN classifiers do not use any model to fit and only based on memory. KNN algorithm used neighborhood classification as the prediction value of the new query instance. Fig 4 shows KNN Algorithm.

In activity recognition, the k-nearest neighbour algorithm (K-NN) is a non-parametric (or distribution free) method for classifying objects based on closest training samples in the feature space. KNN is a type of memory-based learning, where the function is only approximated locally and all computation is delayed until classification. The KNN algorithm is amongst the simplest of all machine learning algorithms: here, an object is classified by a majority vote of its neighbors, with the object being assigned to the class amongst its k nearest neighbors (k is small positive integer). If k=1, then the object is merely assigned to its nearest neighbor class. The training examples are vectors in a multidimensional feature space, each with a labeled class. The training (data pre-processing) phase of the algorithm consists storing the feature vectors and class labels of the training samples. In the classification phase of activity recognition system, k is a user-defined constant, and an unlabeled vector is classified by assigning most frequent label among the k training samples.

Usually Euclidean distance is used as the distance metric for continuous variables;

$$\text{Euclidean} \quad \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Hamming distance is used as distance metric for discrete variables, such as text classification.

$$D_H = \sum_{i=1}^{k} |Xi - Yi|$$

$$X = Y \rightarrow D=0$$

$$X \neq Y \rightarrow D=1$$

The classification accuracy of *K*NN can be improved significantly if the distance metric is learned with specialized algorithms such as Large Margin Nearest Neighboror or Neighbourhood components analysis.

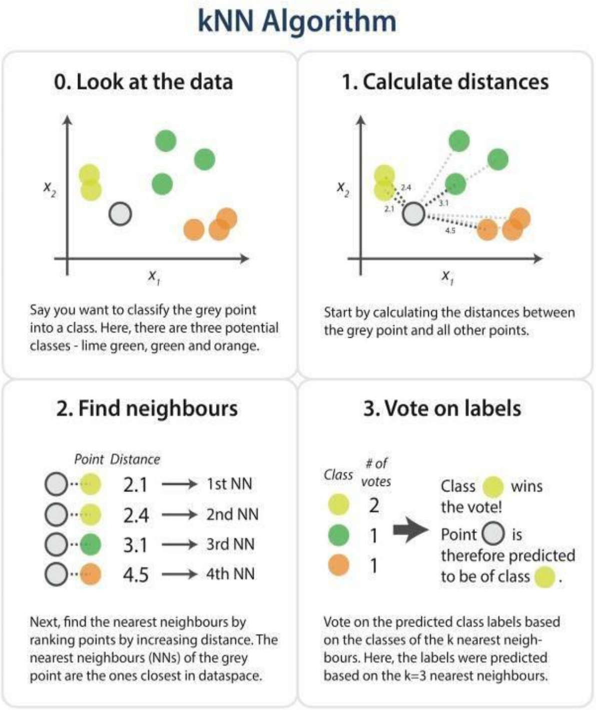

Fig 4 : KNN Algorithm.

2 Clustered KNN

earest Neighbor classifier may have a problem when training samples are uneven. The problem is that KNN classifier may decrease the accuracy of classification because of the uneven density of training data. To solve this problem, a new clustering-based KNN method is presented in this paper. In the first step the training data is pre- processed and four features, which are average, minimum, maximum, and standard deviation, are extracted by using clustering and in the second step classification takes place with a new KNN algorithm; which adopts a dynamic adjustment in each iteration for the neighborhood number K. Clustering is a process of partitioning data into clusters of similar objects. It is an unsupervised learning process of hidden data.

i) Data Pre-processing in Clustered KNN

The main objective of the pre-processing step is to define activity sets from the training data based on the mentioned features. Instead of comparing all the data in the training set, we compare the test data only with the compact training data set that we selected from the original training set. During the pre-processing step, for each feature and for each activity, compact training sets are created. For each feature, except the standard deviation, K data points are selected from the training data. For instance, for the minimum feature set, K ―minimum‖ data points are selected from the training data. Likewise, we create a ―maximum‖ set by selecting the K maximum data points. The average value of the training data is calculated and the nearest K data points are included in the ―average‖ set. For the ―standard deviation‖ set, standard deviation value of training data for each activity is calculated. However, at the same time accuracy of the results are expected to decrease with smaller value of K, so that there is an important trade-off between accuracy and execution time considering the value for K.

ii) Classification in Clustered KNN

In the classification step, we collect test data, in other words we segment the data during a window with a predefined size. After the window is filled, classification phase will starts, and average, minimum, maximum, standard deviation values of the data in the window is calculated and these values are compared one by one with the values in the compact training sets which were created during the pre-processing step. *K* nearest sample to test data is selected from training sets and voting is done by looking at the final list of activities. We label the data in the related window as the activity for which we have maximum amount of data in the final *K* set. The one which is closer to the standard deviation of that particular window is selected as the recognized activity by the standard deviation feature. At the end, we have four labels from voting results of each feature. We label the window as the activity for which we have the highest vote and finalize the classification.

V. HUMAN ACTIVITY RECOGNITION ON ANDROID PLATFORM

This section describes the human activity recognition system based on smartphone, The clustered KNN classifier are implemented on Android phones to detect five main activities; which are walking, running, lying down, standing and sitting. For this purpose, the process is divided into two phases: Data pre-processing and Classification as shown in Fig 5.

In order to monitor the performance of the classifier, the ground truth data is logged, i.e. which activity is literally performed by the user. For this purpose, the application gives voice commands repeatedly to perform an activity. The order of activity is predefined in the system whereas activity duration ―order interval‖ is given directly as the user input to the system in the unit of seconds. During our experiments each activity is performed for 60 sec/cycle. Finally, using these ground truth values, i.e., activity tags, activity recognition performance and other performance metrics of the classifiers are calculated.

Phase I : Data Pre-processing

1) Data collection

Data is collected through the application we created called activity logger and this component is responsible for collecting the training data for each activity separately from sensors, i.e. accelerometer data. In this application, user will selects the activity to be performed, keep the phone into the packet and starts to perform the related activity. For each activity, this application is responsible for creating different training data files in which raw data from the 3-axes of the accelerometer is being logged.

2) Data filtering

The collected data may contain noise and the data collection is processed the data to eliminate the noise by applying noise filters and low-pass filter technique. Time-domain and frequency-domain features have been extensively used to filter relevant information within acceleration and rotation signals. In this paper, we used four features : MIN, MAX, MEAN and Standard deviation. By these statistical operations features were calculated.

3) Data segmentation

Data segmentation is a crucial stage in the activity recognition process; normally sliding window approach is used for segmentation but no clear consent exists on which window size should be preferably employed. Intuitively, decreasing the window size allows faster activity detection, as well as reduced resources and energy needs. On the contrary, large windows are usually considered for the recognition of complex activities. The filtered sensor data is divided into small segments for feature extraction using windowing approaches.

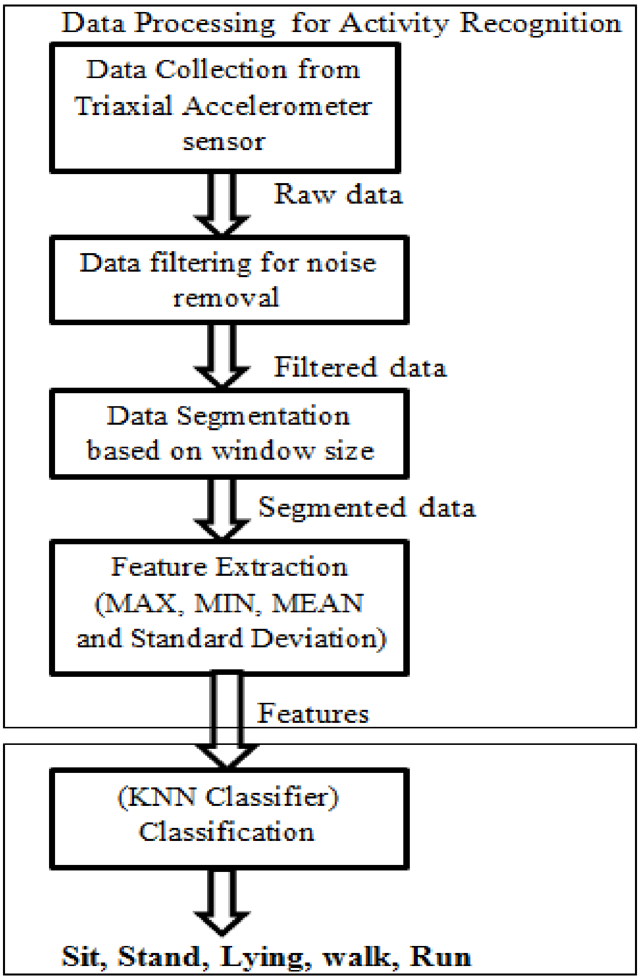

Fig 5 : Human Activity Recognition System

4) Feature Extraction

As in any other data mining tasks, extracting the _right‘ features is critical to the final recognition performance. For activity recognition, we can extract features in both time and frequency domains.

Feature extraction is an most important pre-processing step to activity recognition. It is often decomposed into feature construction and feature selection. When the input data to an algorithm is too large to be processed and it is suspected to be useless then the input data will be transformed into a reduced representation set of features. Transforming the input data into set of features is called feature extraction. If the features extracted are carefully chosen it is expected that the features set will extract the most relevant information from the input data in order to perform the specified task using this reduced representation set of features instead of the full size input. Feature extraction is performed on raw data prior to applying *K*NN algorithm on the transformed data in feature space.

a) Dimensionality Reduction

As shown in Figure 4.1, the Dimensionality Reduction is responsible for reducing the computational complexity thus reducing the response time for recognition process, whereas ensuring accuracy of recognition. It contains two components: the Feature Selection Component and the Instance Selection Component. The Feature Selection is responsible for identifying and removing unneeded, irrelevant and redundant attributes from the dataset. The Instance Selection is responsible for removing unneeded, irrelevant and redundant instances from dataset.

b) Feature Selection

The Feature selection will reduce the processing cost by remove irrelevant and redundant features, whereas ensuring the accuracy of recognition. The redundant features are those which do not contribute information to the recognition process. In other words, the features which provide no/irrelevant to the selected current features set should be eliminated. As a result, it will improve model interpretability, shorter training time, and enhance generalization by reducing over fitting.

Phase II: Classification

When the window is filled, classification phase will starts, and average, minimum, maximum, standard deviation values of the data in the window is calculated and these values are compared one by one with the values in the compact training sets which were created during the pre-processing step. *K* nearest sample to test data is selected from training sets and voting is done by looking at the final list of activities. We label the data in the related window as the activity for which we have maximum amount of data in the final *K* set. The one which is closer to the standard deviation of that particular window is selected as the recognized activity by the standard deviation feature. At the end, we will get four labels from voting results of each feature. We label the window as the activity for which we have the highest vote and finalize the classification.

VI. RESULT AND DISCUSSION

To evaluate the performance of clustered KNN classification for each activity. The confusion matrix for clustered KNN is presented in Table 1. Confusion matrix is a visualization tool typically used in supervised learning techniques. One advantage of confusion matrix is that it is easy to see if the system is confusing two classes. Each column of the confusion matrix represents the instances in a predicted class, while each row represents the instances in an actual class. A confusion matrix contains information about known class labels and predicted class labels. Compared to the performance of activities like running, lying down, standing and sitting, the KNN classifier presents slightly worse performance for walking, where this activity is sometimes classified as running or standing. However, the overall performance for clustered KNN classification is around 92% accuracy considering all activities.

We are also evaluated the impact of *K* value on the performance of clustered KNN classification for each activity. As we expected, increasing the *K* value will affects accuracy rates positively. Whenever we consider the overall effect of all system parameters, we observe best results in the case where K is selected as 50, window size is selected as 1 second and sampling interval is selected as 50 msec.

Table 1: Confusion Matrix

Window size(sec)		0.5			1			2		
Sampling Interval(msec)		10	50	100	10	50	100	10	50	100
K	10	87.9	87.8	87.7	88.4	90.3	89.5	88.6	87.8	89.3
	50	91.1	90.0	91.4	91.9	92.1	90.8	88.9	89.4	91.0

VII. CONCLUSION

In this paper, we proposed an activity recognition system working on Android platforms by developing an application called application logger that supports on-line training and classification while using only the accelerometer data for classification. The performance of on-line classification of KNN classifier is evaluated first then a clustered KNN method is used. The clustered KNN classification exhibit a much better performance than the KNN classifier in terms of accuracy on android platforms with limited resources. We also evaluated the performance of clustered KNN in terms of execution times. As we expected, classification execution times are considerably reduced as K parameter is decreased. Additionally, classification times are highly dependent on the device model and capabilities as well.

Future work will evaluate whether additional features are necessary to improve classifier performance, without adding computational complexity to the algorithm and we will investigate the performance in more complex activities recognition such as bicycling, fall detection.

References

[1] Xing Su, Hanghang Tong, and Ping Ji , –Activity Recognition with Smartphone Sensors‖ ,tsinghua science and technology ISSN ll1007- 0214ll02/11llpp235-249 Volume 19, Number 3, June 2014.

[2] Nicholas D.Lane , Emiliano Miluzzo, Hong Lu, Daniel Peebles,‖A Survey of Mobile Phone Sensing‖,IEEE Communications Magazine September 2010

[3] Sahak Kaghyan, Hakob Sarukhanyan, ‖Activity recognition using knearest neighbor algorithm on smartphone with tri-axial accelerometer ‖,International Journal "Information Models and Analyses" Vol.1 / 2012J.

[4] Mustafa Kose, Ozlem Durmaz Incel,Cem Ersoy ‖Online Human Activity Recognition on Smart Phones‖2nd International Workshop on Mobile Sensing, April 16, 2012.

[5] Pekka Sirrtola and Juha Röning, –Recognizing Human Activities Userindependently on Smartphones Based on Accelerometer Data‖,International Journal of Artificial Intelligence and Interactive Multimedia, Vol. 1, No 5.

[6] Jennifer R.Kwapisz Gary M. Weiss, Samuel A. Moore,‖Activity Recognition using Cell Phone Accelerometers‖,sensor KDD '10,july 25,2010 ,Washington,DC,USA.copyright 2010ACM.

[7] Ling bao and stephen S. Intille,‖Activity Recognition from User-Annotated Acceleration Data‖,. Ferscha and F.Mattern (Eds) PERVASIVE 2004, LNCS 3001, pp. 1–17, 2004. Springer-Verlag Berlin Heidelberg 2004.

[8] Won-Jae Yi, Weidi Jia, and Jafar Saniie ,‖Mobile Sensor Data Collector using Android smartphone‖, Department of Electrical and computer Engineering illinos institute of technology 3301 S.Dearborn St.103SH,Chicago IL,USA.

[9] I. Anderson, J. Maitland, S. Sherwood, L. Barkhuus, M. Chalmers, M. Hall, B. Brown, and H. Muller. Shakra: tracking and sharing daily activity levels with unaugmented mobile phones. Mobile Networks and Applications, 12(2-3):185–199, 2007.

[10] Android. Sensors Overview. http:// developer.android.com/ guide/ topics/ sensors/ sensors_overview.html, 2014. [Online; accessed 01-March-2014].

[11] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In Ambient Assisted Living and Home Care, pages 216–223. Springer, 2012.

[12] Apple. Nike + iPod Application, 2014. [Online; accessed 01-March-2014].

[13] Apple. UIAcceleration Class Reference. https://developer.apple.com/library/ios/documentation/uikit/reference/UIAcceleration_Class/Reference/UIAcceleration.html, 2014. [Online; accessed 17-March-2014].

[14] W.-Y. Deng, Q.-H. Zheng, and Z.-M. Wang. Cross-person activity recognition using reduced kernel extreme learning machine. Neural Network

\