Activity Recognition Using Smartphone Sensors

Alvina Anjum and Muhammad U. Ilyas
Applied Network & Data Science Research Group (AN-DASH)
School of Electrical Engineering & Computer Science (SEECS)
National University of Sciences & Technology (NUST)
Islamabad – 44000, Pakistan
{10mscseaanjum, usman.ilyas}@seecs.edu.pk

***Abstract*—Motion sensor embedded smartphones have provided a new platform for activity inference. These sensors, initially used for cell phone feature enhancement, are now being used for a variety of applications. Providing cell phone users information about their own physical activity in an understandable format can enable users to make more informed and healthier lifestyle choices. In this work, we built a smartphone application which tracks users' physical activities and provide feedback requiring no user input during routine operation. The application reports estimates of the calories burned, broken up by physical activities. Detectable physical activities include walking, running, climbing stairs, descending stairs, driving, cycling and being inactive. We evaluated a number of classification algorithms from the area of Machine Learning, including Naïve Bayes, Decision Tree, K-Nearest Neighbor and Support Vector Machine classifiers. For training and verification of classifiers, we collected a dataset of 510 activity traces using cell phone sensors. We developed a smartphone app that performs activity recognition that does not require any user intervention. The classifier implemented in the Android app performs at an average true positives rate of greater than 95%, false positives rate of less than 1.5% and an ROC area of greater than 98%.**

I. Introduction

A. Motivation & Problem Statement

The last decade witnessed a virtual explosion of mobile communication infrastructure and services. With the technological advancement of cell phones a new platform for computing is quickly gaining popularity. These state of the art mobile phones are becoming more popular than computers as they wireless computers of the new era. The current smart phones are not just communication devices rather it's a personal computer packed into a small gadget. Smartphones are now equipped with a number of embedded sensing devices such as an accelerometer, gyroscope, digital compass, microphone, GPS and camera. These sensors are now being used in various fields for human gesture and activity recognition based applications which are opening the doors to new areas of research and significantly impacting our daily life. Thus, the recognition of human physical activities from cell phone sensor measurements is formulated as a detection problem.

B. Limitations of Prior Art

The explosive growth in the global penetration rate of cell phones and, more recently, smartphones has led to interest in leveraging this connectivity for the delivery of health care services to the most remote regions. One of the many health care applications that cell phones are being leveraged for is activity recognition. Prior works on automated activity recognition, like Bao and Intille [1], required the use of dedicated on-body sensors. Prior approaches that rely only on cell phone sensors such as ours either required calibration or require users to keep their phones in a particular way. Kwapisz, Weiss and Moore [2] require the phone to be in a pant pocket. A recent study by Lane *et al.* [3] implements jigsaw continuous sensing [4] to develop an application based on physical health, sleep and emotional well being. The physical activity section of this application focuses only on three main activities walking, sitting and running. Another drawback of the system is that the report of the activity classification can only be viewed on a desktop web portal. A basic level mobile application uses graphical user interface to display physical well being along with a score which is computed using Metabolic Equivalent of Task (MET). Classification is performed using a split and merge technique [4]. Although classification accuracy of more than 90% is achieved, the downside of using the split and merge method is that the classifier is designed offline and merged at runtime. Another drawback is the development of multiple classifiers for various activity subclasses which adds to the classifier's complexity. Most apps available on app marketplaces today that seek to track calorie consumption require extensive user input.

C. Proposed Approach

Our proposed approach is to develop a classifier while adhering to best machine learning practices. We collect a large data set for classifier training and verification. We target the recognition of 7 different activities, which include walking, running, climbing stairs, descending stairs, cycling, driving and remaining inactive. Cell phone physical sensor readings were collected while performing all these activities, while the cell phone was kept in different orientations. The data set so collected was pre-processed to extract a number of features from the data that have not been explored in previous studies. The most informative features were chosen for further use by ranking features according to information gain. The data set was partitioned into 10 sets for 10-fold cross validation. Various classification algorithms were trained using the training set and then evaluated using the test set, iterating over all 10 data sets. The best performing classifier was chosen for implementation in the Android app.

978-1-4673-5133-3/13/$31.00 ©2013 IEEE

D. Experimental Results & Findings

We designed four separate classifiers, namely C4.5 Decision Tree, Naïve Bayes, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). For verification, 10-fold cross validation was carried out. It was determined that the C4.5 Decision Tree classifier outperforms the other classifiers on average with a true positive rate of 95.2%, false positive rate of 1.1%, precision of 94.4% and recall rate of 94.2%.

E. Key Contributions

Our contributions are three-fold:

1) We created a data set comprising 510 activity traces. These comprise of measurements taken from cell phone physical sensors collected while their carriers were engaged in various physical activities.
2) We developed four different classifiers for the task of activity detection, the best of which provides excellent average performance in terms of hit, false alarm, precision and recall rates.
3) We developed an Android app that implements the classifier and performs live activity detection and reporting.

Paper Organization: The remainder of this paper is organized as follows. Section II describes some of the related work in the area of physical activity detection. Section III describes details of the data set collected for this study. Section IV describes the pre-processing, feature extraction and training of classification algorithm. Section V reports detailed performance results of the classification algorithm. Section VI describes the finished Android app. Section VII concludes the paper.

II. Related Work

Activity recognition has a number of leading application areas. Wearable accelerometer based body sensors have been used to infer human mobility levels. These sensors have favorable have small size and high level of accuracy. Earlier works explored the use of multiple on-body sensors placed at the waist, arms, knees and ankles [1] to accurately determine physical activities ([5]–[8]). Other works investigated activity recognition outside laboratory conditions but lacked accuracy ([9], [10]). The positioning of sensors was crucial for precise inference, e.g. eating, typing and brushing teeth were easier to determine if placed at the wrist and arm with an accuracy of 96.67%. However, if the same activity is performed with sensors on waist or knee, detection accuracy drops to 66% ([6], [10]). Other disadvantages of using multiple accelerometers include discomfort of numerous wires attached to the body as well as the irritability that comes from wearing sensors for a long duration.

A few works have been performed whereby only one triaxial accelerometer located at the waist or back is used for movement recognition. However accuracy drops to 35% in comparison to 5 on-body sensors ([7], [8], [11]). Nevertheless using single sensor is more preferable and convenient in real world scenarios. These sensors were capable of recognizing basic activities such as lying, walking and running. Reduction in sensors requires analysis of fewer signals thus reducing computational power. Restricting the user to or adopt a particular lifestyle will result in a low adoption / acceptance rate. Recent studies have shown the use of a single device fitted with multiple sensors on a Multi-modal Sensor Board (MSB) ([12], [13]). Lester et al. used a MSB for activity recognition and a Bluetooth device to send the data to a hand held device or laptop for offline analysis [14]. On the downside these devices have limited battery power.

Accuracy rates of up to 91% have been reported if phones are allowed to be carefully positioned in pre-determined locations ([1], [2], [12], [15]). Sun et al. [16] investigated orientation and position independent classification schemes whereby a separate classifier is implemented for each position. This approach yielded an accuracy of 93% but is not practically feasible as it requires high computational resources. Some recent studies show behavioral analysis based applications using activity recognition techniques. In [17] the fluctuations in the signal strength of GSM signals are used to infer activities like sitting, walking and driving. The use of GSM signals is not a very efficient approach as there are sudden spikes and troughs that cause complexity in GSM cell strength and visibility. Real time activity inferences also leads to accuracy and privacy issues. Recently, smartphone GPS sensor data was also exploited to establish mode of transport used to commute over duration of time. Consolvo et al. [18] used a mobile sensing platform called UbiFit for activity recognition which transfers data to a Nokia smartphone which uses blooming flowers to encourage increase in physical activity. Similarly, Denning et al. [19] worked on BALANCE which estimates the calorie expenditure in routine life. Nevertheless both these solutions rely on additional body sensors.

In the last few years attempts have been made to implement activity recognition for limited sets of activities using mobile phones. The CenceMe system developed by Miluzzo *et al.* [20] explored basic level activities like idle walking and running for continuous mobile sensing with the help of back-end classifiers using accelerometer data along with conversation data. This approach achieved an overall classification accuracy of 72%. Hausmann [21] classified idle, walking and running into activity levels using median thresholds which are validated by empirical experimentation (rather than using Machine Learning algorithms).

III. Data Set

The development of the proposed activity detection classifier will proceed using a supervised learning algorithm. This requires a data set comprising of a large number of cell phone sensor measurements collected during all targeted physical activities. To this end, we developed a separate cell phone app for the purpose of collecting such a training data set (see screenshots in Figure 1). Like the main app, the data collection app is also developed for Google's Android operating system. Data collection was performed on a Samsung Galaxy Y phone with Android Gingerbread version 2.3.3. The choice of a low-cost smartphone was deliberate to demonstrate the viability

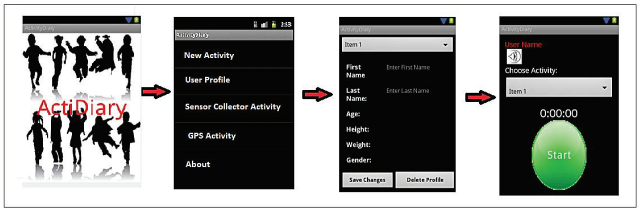

Fig. 1: Screenshots of Data Collection app.

of performing activity recognition on even the most resource constrained smartphones.

The primary considerations in the design of the data collection app were simplicity and ease-of-use. The first time it is used, users are asked to complete a short, simple profile. This profile requires basic information like name, age, height, weight, gender and most common placement of the phone. The primary function of the app is to let users a) specify what physical activity they are about to perform, b) start sensor data collection and c) let users stop data collection after the physical activity has been completed (in this order). The app lets users select one of the targeted physical activities, i.e. inactive, walking, running, climbing stairs, descending stairs, cycling and driving. The sensor data logged from each activity is written into a comma separated values file called an *activity trace*. Phone placements during data collection were varied and included placement in hand, pants pocket, shirt pocket and handbag. Each activity trace contains data consisting of time series of 3 accelerometers ($a_x[n]$, $a_y[n]$, $a_z[n]$) and 3 gyroscopes ($g_x[n]$, $g_y[n]$, $g_z[n]$). It also contains data from the phones GPS unit that includes longitude, latitude and speed. Table I shows the number of activity traces collected for various activities, broken up by the phone placement.

Android permits three different modes with different sampling rates that can be used to collect data from accelerometer sensors. These are called 'Normal' ($5Hz$), 'UI' ($15Hz$), 'Game' ($50Hz$) and 'Fastest' (platform dependent, $\geq$ 'Game' and up to $100Hz$). The data collection app uses the 'UI' mode because a Nyquist rate of $15 - 16Hz$ allows a maximum signal frequency of $8Hz$ which was found to be adequate for human physical activity. Increasing sensor sampling rates beyond this level incurs an unnecessary penalty in device power consumption. We collected data from ten different people ranging in age from as little as 12 to 25. A group of four people were used for training the data and the remaining users provided test data for the activity classification.

TABLE I: Data set of activity traces.

Activity \ Placement	Pant Pocket	Hand	Hand Bag	Shirt Pocket	Sub-total
Walking	25	20	15	20	**80**
Running	20	20	25	15	**80**
Climbing Stairs	15	25	15	10	**75**
Descending Stairs	15	20	15	10	**60**
Driving	10	10	20	25	**65**
Cycling	25	15	15	10	**65**
Inactive	20	20	25	20	**85**
TOTAL					**510**

IV. Classifier Design

A. Pre-processing

For the process of designing activity classifiers we followed standard machine learning practices for supervised learning algorithms. Note that the x, y and z axes of the accelerometer and gyroscope sensors are defined relative to the body of the phone. The varying orientation of a phone does not allow a meaningful comparison of measurements of a particular axis' accelerometer with measurements of the same axis accelerometer from a different activity trace. Therefore, we rotate the three orthogonal reference axes to align with the three axes d_1, d_2 and d_3 corresponding to the axes descending order of signal variation. This is done using Eigen-decomposition of the covariance matrix of a_x, a_y and a_z. The sample covariance of any two axis i and j is computed using samples in activity trace files as shown in Equation 1, where N denotes the number of samples of each sensor and $\overline{a}_i$ denotes the mean of a_i.

$$\sigma_{ij} = \frac{1}{N-1} \sum_{n=1}^{N} \left(a_i[n] - \overline{a}_i\right)\left(a_j[n] - \overline{a}_j\right) \tag{1}$$

The covariance matrix $\mathbf{C}$ is defined in terms of covariance terms as,

$$\mathbf{C} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix} \tag{2}$$

If $\mathbf{V}$ is the matrix of eigenvectors, then the matrix $\mathbf{A}$ of accelerometer signals is transformed into matrix $\mathbf{D}$ according to Equation 3.

$$\begin{aligned} &\mathbf{D} = \mathbf{AV} \\ &\text{where} \\ &\mathbf{D} = \begin{bmatrix} d_1[n] & d_2[n] & d_3[n] \end{bmatrix} \\ &\mathbf{A} = \begin{bmatrix} a_x[n] & a_y[n] & a_z[n] \end{bmatrix} \end{aligned} \tag{3}$$

B. Feature Extraction

The finished application is designed to refresh its estimate of physical activity every 5 seconds. Therefore, feature extraction from the activity traces constituting the training data is performed from matching time windows of 5 seconds. [1] Previous studies used features from both time and frequency domain representations of signals for offline activity recognition. Features computed from time domain signals that have been used previously include mean, standard deviation, magnitude of acceleration, cross-axis signals correlation. Frequency domain features include Fast Fourier Transform (FFT) spectral energy, frequency domain entropy, log of FFT etc.

[1]The first 2 seconds of data from every activity trace are discarded to avoid signal noise resulting from the placement of the cell phone after tapping the data collection app's 'Start' button.

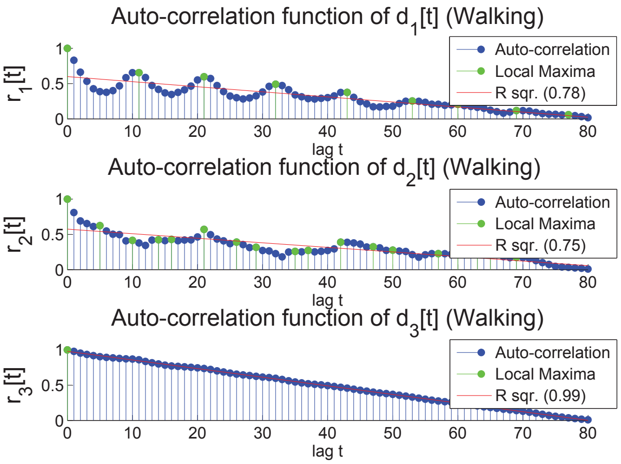

Fig. 2: Auto-correlation function of a coordinate transformed activity trace collected while walking, highlighting local maximas and an optimally fitted linear model $f(t)$.

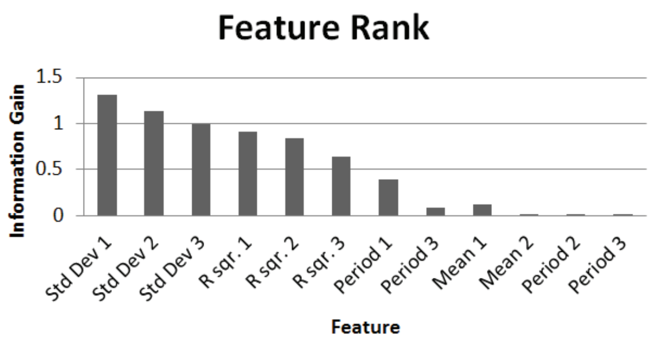

Fig. 3: Feature rank by information gain.

Relying on features derived solely from the time domain signal representation is computationally less expensive because it avoids computation of FFT. Exploratory analysis of the three transformed accelerometer signals $\mathbf{D}$ showed that the Probability Density Functions (PDF) of the samples comprising $d_i[n]$, where $1 \leq i \leq 3$, are Gaussian. Furthermore, we also computed the autocorrelation function of all accelerometer signals and included its features to our feature space. The correlation coefficient function of $d_i[n]$ where $1 \leq i \leq 3$ is defined in Equation 4.

$$r_i[t] = \frac{1}{N-1} \sum_{n=1}^{N} \frac{\left(d_i[n] - \overline{d_i}\right)\left(d_i[n+t] - \overline{d_i}\right)}{\sigma_i^2} \tag{4}$$

Mean: The means of all transformed accelerometer signals over 5 sec intervals: $\overline{d}_i$ where $1 \leq i \leq 3$. These, as it would later turn out would not prove very useful. Therefore, we also computed the means of the autocorrelation function: $\overline{r}_i$ where $1 \leq i \leq 3$.

Variance: The variances of all transformed accelerometer signals as well as the autocorrelation function computed over 5 sec windows: σ_{di}^2 and σ_{ri}^2 where $1 \leq i \leq 3$.

Period: Most physical activities whose detection is supported by the app are periodic and involve repetitive motion. We expected this to be reflected in the signal along some axis of the accelerometer signals, and expected the frequency of the person's motion to match the dominant frequency of those signals. A first attempt was made to isolate this dominant frequency by analyzing signals' FFT coefficients. However, it became immediately clear there is no discernible difference between the frequency spectra of signals with or without obviously identifiable period in the time domain. We explain this failure by the noise in the time domain signal. Therefore, we adopted the step-wise approach based on the correlation coefficient function. Operating on the autocorrelation function serves to denoise the signal.

The period of the signal is estimated from the correlation coefficient function $r_i[t]$ by the following steps.

1) Identify samples that are local maximas.
2) Compute the time difference (in number of samples) between successive maximas.
3) Estimate the period of the signal as the median inter-maxima delay. Its inverse is the frequency.

Figure 2 are three stem plots of the auto-correlation functions of $r_1[t]$, $r_2[t]$ and $r_3[t]$ of $d_1[n]$, $d_2[n]$ and $d_3[n]$, respectively, for an activity trace in which the cell phone user is walking. The green stems among the mostly blue stems are the ones that are identified as local maximas by the three-step method described above.

Goodness of Linear Fit: The goodness of linear fit, denoted R^2, of a linear model for the correlation coefficient function $r_i[n]$ of $d_i[n]$, where $1 \leq i \leq 3$, is a measure of the variability of the data. The R^2 goodness of fit measure is computed according to Equation 5.

$$R^2 = 1 - \frac{S_e}{S_t} \tag{5}$$

where

$$S_t = \sum_{n=1}^{N} \left(r[n] - \overline{r}\right)^2 \text{and} S_e \quad = \sum_{n=1}^{N} \left(r[n] - f[n]\right)^2 \tag{6}$$

We observed that the R^2 measure of activity traces with low levels of activity is higher compared to the R^2 of the corresponding signal in activity traces with high-levels of activity. In Figure 2, the red line is the optimum linear fit $f(t)$ for the auto-correlation functions $r_i[t]$ where $1 \leq i \leq 3$. It is used in Equation 6 to compute S_t and S_e which are in turn used to compute the R^2 goodness of fit in Equation 5.

C. Classifier Training & Validation Strategy

We trained and evaluated four different classification algorithms: Navïve Bayes, C4.5 Decision Tree, K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). We used information gain as a measure to rank all features according to their usefulness to classification. We discovered that features derived from the transformed accelerometer signals ($d_1[n]$,

$d_2[n]$ and $d_3[n]$) ranked significantly lower than features derived from their autocorrelation functions. Similarly, the gyroscope sensor measurements were also found to be of no value. This confirms the findings by Brush et al. [22] who tried to use gyroscope measurements in addition to accelerometer measurements. Therefore, the means and variances/standard deviations of the accelerometer signals and all features derived from the gyroscope sensor measurements were removed from further consideration. The ranking of remaining features by information gain is shown in Figure 3. Based on this ranking, only the top 9 features were used for classification. The KNN classifier permits for different values of K in its use. Each value of K produces, in effect, a separate classifier. Following some exploratory analysis, we determined that increasing the value of K beyond 1 leads to a consistent decrease in performance metrics. Therefore, the performance numbers given in the subsequent section are for KNN with $K = 1$.

For performance evaluation we used 10-fold cross validation. This means, activity traces in the data set were divided into 10 sets. One set was selected to serve as test data, while the remaining 9 were used for training the classifier. This procedure was performed 10 times while iterating the test set over the 10 partitions of the data set.

TABLE II: Detailed Accuracy By Class (Naïve Bayes).

Class	TP Rate	FP Rate	Precision	Recall	ROC Area
Walking	0.801	0.082	0.784	0.801	0.952
Running	1.000	0.009	0.935	1.000	1.000
Climbing Stairs	0.616	0.054	0.707	0.616	0.933
Descending Stairs	0.868	0.03	0.817	0.868	0.973
Driving	0.914	0.024	0.881	0.914	0.982
Cycling	1.000	0.002	0.950	1.000	1.000
Inactive	0.925	0.000	1.000	0.925	0.994
Weighted Average	**0.847**	**0.0401**	**0.838**	**0.84**	**0.968**

TABLE III: Detailed Accuracy By Class (C4.5 Decision Tree).

Class	TP Rate	FP Rate	Precision	Recall	ROC Area
Walking	0.978	0.049	0.890	0.978	0.976
Running	1.000	0.002	1.000	1.000	1.000
Climbing Stairs	0.837	0.01	0.846	0.837	0.989
Descending Stairs	0.925	0.007	0.905	0.925	0.978
Driving	1.000	0.007	1.000	1.000	1.000
Cycling	1.000	0.000	1.000	1.000	1.000
Inactive	0.943	0.000	1.000	1.943	0.978
Weighted Avg	**0.952**	**0.011**	**0.944**	**0.942**	**0.985**

TABLE IV: Confusion matrix (Decision Tree).

a	b	c	d	e	f	g	← Classified As Actual Activity ↓
133	0	0	2	1	0	0	a ← Walking
0	**57**	1	0	0	0	0	b ← Running
13	0	**72**	3	0	0	0	c ← Climbing Stairs
3	0	0	**62**	1	1	0	d ← Descending Stairs
0	0	0	0	**78**	0	0	e ← Driving
0	0	0	0	0	**19**	0	f ← Cycling
0	1	1	0	1	0	**50**	g ← Inactive

V. Performance Evaluation

In this section we report detailed results of the performance analysis we conducted for all four types of classifiers. The Naïve Bayes classifier treats all features as independent and is by far the simplest of these four classifiers. For all classifiers, we report True Positives (TP) rate, False Positives (FP) rate, Precision, Recall rate and ROC curve area. Table II shows the performance of the Naïve Bayes classifier in terms of TP rate, FP rate, precision, recall rate and ROC curve area for each activity. Among the classifiers we trained Naïve Bayes is the only one that makes the assumption that all features are independent. The last row in Table II consists of the weighted averages of these metrics (weighted by the number of examples in each class). With the exception of 'climbing stairs', the FPs of all other activities are below 20% with most all other rates above 80%.

Table III is the same table of performance metrics broken up by activity for the C4.5 Decision Tree classifier. Its performance is significantly better than that of the Naïve Bayes classifier and matches or beats the performance of the subsequently evaluated KNN and SVM classifiers. All FPs are below 5% with all other rates ranging between 89% and 98%. Due to brevity constraints, and due to the fact that the C4.5 Decision Tree classifier yielded the best performance, we provide the confusion matrix for only the Decision Tree classifier in Table IV. Subsequent Tables V and VI are tables of performance metrics for the KNN with $K = 1$ and SVM classifiers. Both are clearly outperformed by the C4.5 Decision Tree on all performance metrics.

TABLE V: Detailed Accuracy By Class (K-Nearest Neighbor).

Class	TP Rate	FP Rate	Precision	Recall	ROC Area
Walking	0.789	0.042	0.544	0.789	0.776
Running	1.000	0.000	1.000	1.000	1.000
Climbing Stairs	0.705	0.020	0.585	0.705	0.812
Descending Stairs	0.905	0.018	0.792	0.905	0.932
Driving	0.832	0.018	0.714	0.832	0.776
Cycling	0.824	0.000	1.000	1.000	1.000
Inactive	0.842	0.000	1.000	1.000	1.000
Weighted Avg	**0.887**	**0.011**	**0.805**	**0.887**	**0.894**

TABLE VI: Detailed Accuracy By Class (Support Vector Machine).

Class	TP Rate	FP Rate	Precision	Recall	ROC Area
Walking	0.897	0.275	0.550	0.897	0.834
Running	0.966	0.002	0.982	9.996	0.998
Climbing Stairs	0.442	0.027	0.929	0.422	0.898
Descending Stairs	0.388	0.005	0.776	0.388	0.902
Driving	0.778	0.031	0.829	0.778	0.943
Cycling	1.000	0.004	0.905	1.000	0.998
Inactive	0.894	0.004	0.957	0.894	0.968
Weighted Avg	**0.738**	**0.086**	**0.791**	**0.738**	**0.911**

VI. Activity Diary Android App

One of the key contributions of our work is the implementation of a smartphone app that performs live activity detection of the person carrying it without human intervention. A primary design consideration has been minimization of user input in the working of the app. The Activity Diary app provides users a report of their physical activity level

Fig. 4: Screenshots of Activity Diary app.

(screenshots in Figure 4). Using the personal information provided by users, the app also estimates the calories expended in the course of these physical activities. The activity levels for each activity are displayed on the screen as shown in the figure of the application screenshots. Information about the duration for which an activity is performed is used to calculate the calories burned. For personal interpretive reference, the app also computes each user's Basal Metabolic Rate (BMR). The BMR is the number of calories that can be consumed without any physical activity to maintain the same weight. The BMRs for women B_W and men B_M are computed separately in Equation 7 using weight (lbs), height (in) and age (yrs).

$$\begin{aligned} B_W &= 655 + 4.35 \times \text{wt.} + 4.7 \times \text{ht.} - 4.7 \times \text{age}, \\ B_M &= 66 + 6.23 \times \text{wt.} + 12.7 \times \text{ht.} - 6.8 \times \text{age} \end{aligned} \tag{7}$$

Metabolic Equivalent of Task (MET) is used to measure energy expenditure. The MET are employed to calculate the calories burned during a period of 24 hours. The progress of the user in terms of physical activities and calories burned are communicated to the user by means of graphical reports.

VII. Conclusions

In this paper we reported the design, development and performance evaluation of a smartphone app that performs live detection of physical activities. This app differentiates itself from previous works on activity recognition in the following: 1) It requires no user interaction post-setup, 2) It requires no additional sensing hardware and relies solely on the physical sensors that are standard on even low-end smartphones, 3) It requires no calibration, 4) It supports the detection of 7 different physical activities, including walking, running, climbing stairs, descending stairs, cycling, driving and remaining inactive, and 5)The Decision Tree classifier used by the Activity Diary app is accurate. The average area under the ROC curve exceeds 0.99. This application lets users monitor their daily physical activity and enables them to make healthier and more informed choices that can lead to healthier habits and lifestyle. Live updates are specifically targeted to encourage decisions based on a healthier lifestyle. We are currently field testing the Activity Diary app and are working to make it available to the public through the Google Android marketplace. In the future, we would like to extend this app further to: 1) Support more physical activities, 2) Infer modes of transportation and provide users with feedback in terms of their estimated carbon footprint, and 3) Develop an app function that uses historical information about physical activities and contextual information to gives users pro-active suggestions for lifestyle choices.

References

[1] L. Bao and S. Intille, "Activity recognition from user-annotated acceleration data," *Pervasive Computing*, pp. 1–17, 2004.

[2] J. Kwapisz, G. Weiss, and S. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[3] N. Lane, M. Mohammod, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell, "Bewell: A smartphone application to monitor, model and promote wellbeing."

[4] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Conf. on Embedded Networked Sensor Sys.* ACM, 2010, pp. 71–84.

[5] G. Chambers, S. Venkatesh, G. West, and H. Bui, "Hierarchical recognition of intentional human gestures for sports video annotation," in *Intl. Conf. on Pattern Recognition*, vol. 2. IEEE, 2002, pp. 1082–1085.

[6] S. Lee and K. Mase, "Activity and location recognition using wearable sensors," *Pervasive Computing, IEEE*, vol. 1, no. 3, pp. 24–32, 2002.

[7] P. Lukowicz, H. Junker, M. Staeger, T. von Bueren, and G. Troester, "Wearnet: A distributed multi-sensor system for context aware wearables," *Ubiquitous Comp.*, pp. 361–370, 2002.

[8] M. Mathie, A. Coster, N. Lovell, B. Celler, S. Lord, and A. Tiedemann, "A pilot study of long-term monitoring of human movements in the home using accelerometry," *Journal of telemedicine and telecare*, vol. 10, no. 3, pp. 144–151, 2004.

[9] J. Bussmann, W. Martens, J. Tulen, F. Schasfoort, H. van den Berg-Emons, and H. Stam, "Measuring daily behavior using ambulatory accelerometry: the activity monitor," *Behavior Research Methods*, vol. 33, no. 3, pp. 349–356, 2001.

[10] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[11] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *Info. Tech. in Biomedicine, IEEE Trans. on*, vol. 10, no. 1, pp. 156–167, 2006.

[12] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Intl. workshop on Interactive multimedia for consumer electronics*. ACM, 2009, pp. 1–10.

[13] S. Lau and K. David, "Movement recognition using the accelerometer in smartphones," in *Future Network and Mobile Summit*. IEEE, 2010, pp. 1–9.

[14] J. Lester, T. Choudhury, G. Borriello, S. Consolvo, J. Landay, K. Everitt, and I. Smith, "Sensing and modeling activities to support physical fitness," in *Ubicomp Workshop: Monitoring, Measuring, and Motivating Exercise: Ubiquitous Computing to Support Fitness*, Tokyo, Japan, 2005.

[15] T. Brezmes, J. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," *Distributed Comp., Artificial Intelligence, Bioinformatics, Soft Comp., and Ambient Assisted Living*, pp. 796–799, 2009.

[16] L. Sun, D. Zhang, B. Li, B. Guo, and S. Li, "Activity recognition on an accelerometer embedded mobile phone with varying positions and orientations," *Ubiquitous Intelligence and Comp.*, pp. 548–562, 2010.

[17] 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*, vol. 12, no. 2, pp. 185–199, 2007.

[18] S. Consolvo, D. McDonald, T. Toscos, M. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby *et al.*, "Activity sensing in the wild: a field trial of ubifit garden," in *SIGCHI conf. on Human factors in comp. sys.* ACM, 2008, pp. 1797–1806.

[19] T. Denning, A. Andrew, R. Chaudhri, C. Hartung, J. Lester, G. Borriello, and G. Duncan, "Balance: towards a usable pervasive wellness application with accurate activity inference," in *workshop on Mob. Comp. Sys. and Applications*. ACM, 2009, p. 5.

[20] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Conf. on Embedded network sensor sys.* ACM, 2008, pp. 337–350.

[21] J. Hausmann, "Activity level estimator," Master's thesis, University of Geneva, Switzerland, 2010.

[22] A. Brush, A. Karlson, J. Scott, R. Sarin, A. Jacobs, B. Bond, O. Murillo, G. Hunt, M. Sinclair, K. Hammil *et al.*, "User experiences with activity-based navigation on mobile devices," in *Intl. Conf. on Human computer interaction with mobile devices and services*. ACM, 2010, pp. 73–82.