Project no: 689043
Acronym : SELFBACK
Title: A decision support system for self-management of low back pain
Activity: PHC-28-2015 Predictive modelling RIA

Work Package 2:	Predictive Monitoring Analytics
Deliverable D2.1:	Physical Activity Recognition
Organisation name of deliverable lead:	Robert Gordon University
Author(s):	Sadiq Sani, Nirmalie Wiratunga, Stewart Massie, Kay Cooper
Reviewer(s):	Kerstin Bach, Agnar Aamodt
Participants:	RGU, NTNU
Type:	Demonstrator
Dissemination Level:	PU
Version:	1.1
Total no of pages:	21
Project Start date:	1. January 2016
Contractual delivery date:	30. June 2016
Actual delivery date:	
Keywords:	Activity Recognition, Time Series Analysis
Status:	Approved by the EC

Abstract

This deliverable covers the provision of an activity recognition system from accelerometer data. The system takes as input time series data from a triaxial accelerometer, along with time stamps for each accelerometer reading, and returns a set of detected activities along with start and end times of each detected activity. The system also returns step counts for ambulation activities i.e. walking, climbing stairs and running.

This project has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement No 689043.

Document History

Version	Date	Author(s)	Description
0.1	01/06/16	Sadiq Sani, Nirmalie Wiratunga, Stewart Massie	Initial version of the document
0.2	28/06/16	Sadiq Sani, Nirmalie Wiratunga, Stewart Massie	Revised version of the document after initial review
1.0	30/06/16	Kerstin Bach	D2.1 submitted to the EC
1.1	03/08/16	Kerstin Bach	Updated author list
1.1	28/2/17		Approved by the EC

Table of Contents

1 Introduction

Automated human physical activity recognition is critical to the SELFBACK project. This is needed in order to determine how well non-specific Low Back Pain (LBP) patients adhere to prescribed guidelines for daily physical activity. Accurate reporting of a patient's daily physical activity allows the SELFBACK system to provide useful feedback to help patients better adhere to prescribed guidelines.

Physical activity recognition is the task of automatically inferring human activity from recorded sensor input. For the purpose of this deliverable, the sensor input used is a tri-axial accelerometer mounted on a person's wrist. The output is a string representation of a list of JSON objects for the different detected activities. Each JSON object provides the name of the predicted activity, the start time, the end time, and for ambulation activities (e.g. walking and running), the step count. An overview of this is shown in Figure 1-1.

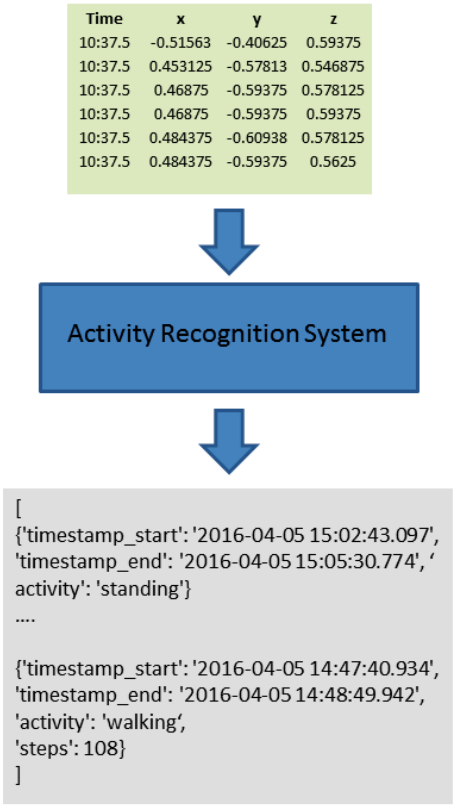

Figure 1-1 Activity Recognition System Overview

The activity recognition system is developed in Python 2.7 and is dependent on Pandas, Numpy, Scikit and Scipy Python libraries. Accordingly, all these libraries need to be installed for the application to run successfully. All these libraries are open source (under the BSD

license) and are available for free download. The code for the SELFBACK activity recognition is also is available for free download on Github[1].

The following sections of this report describe in detail, the SELFBACK activity recognition system. Physical activity recognition is modelled as a supervised machine learning task within the system.
Supervised machine learning algorithms are able to generate accurate prediction models by learning from example data. Hence, labelled sensor data is required for training the supervised learning algorithm. Section 2 describes the data collection approach adopted for the SELFBACK system. The SELFBACK activity recognition algorithm is described in Section 3. This includes descriptions of the feature extraction algorithm and supervised learning algorithm adopted, as well as the algorithms used for computing step counts. Evaluation of the system is described in Section 4 along with reports of activity recognition accuracy achieved.

[1] https://github.com/selfback/activity-recognition

2 Data Collection

Training data is required in order to train the activity recognition system. A group of 52 volunteer participants was used for data collection. All volunteers were either students or staff of Robert Gordon University. The age range of participants is 18 – 54 years and the gender distribution is 52% Female and 48% Male. Data collection concentrated on the activities provided in Table 1

Table 1 Activities used for collecting accelerometer data

Activity Label	Description
Walking Slow	Walking at self-selected slow pace
Walking Normal	Walking at self-selected normal pace
Walking Fast	Walking at self-selected fast pace
Jogging	Jogging on a treadmill and self-selected speed
Up Stairs	Walking up a flight of stairs
Down Stairs	Walking down a flight of stairs
Standing	Standing relatively still
Sitting	Sitting still with hands either on the desk or rested at the side
Lying	Lying down relatively still on a plinth

This set of activities was chosen because it represents the range of normal daily activities typically performed by most people. In addition, three different walking speeds (slow, normal & fast) were included in order to have a better estimate of the intensity of the activities performed by the user. Data was collected using the Axivity Ax3 tri-axial accelerometer[2] at a sampling rate of 100Hz. Accelerometers were mounted on the wrists of participants using specially designed wristbands provided by Axivity. Participants were provided with scripts, which contained related activities e.g. sitting and lying. The scripts guided participants on what activity they should do, how long they should spend on each activity (average of 3 minutes) and any specific details on how they should perform the activity e.g. sit with your arms on the desk. An example script is show in Figure 2-1.

[2] http://axivity.com/product/ax3

Protocol 1 – Walking Activities

NB: For each activity aim for at least 3 minutes duration.

1. Put on wristband
2. Clap 3 x
3. Walk on level surface at self-selected slow walking speed
4. Clap 3 x
5. Walk on level surface at self-selected moderate walking speed
6. Clap 3 x
7. Walk on level surface at self-selected fast walking speed
8. Clap 3 x
9. remove wristband

Figure 2-1 Sample Activity Script

Three claps are used to indicate the start and end of each activity. The three claps produce distinct spikes in the accelerometer signal which make it easy to detect the start and end of different activities in the data. This helps to simplify the annotation of the accelerometer data, by making it easy to isolate the sections of the data that correspond to specific activities. This allows the sections to be easily extracted and aligned with the correct activity label from the script as shown in Figure 2-2.

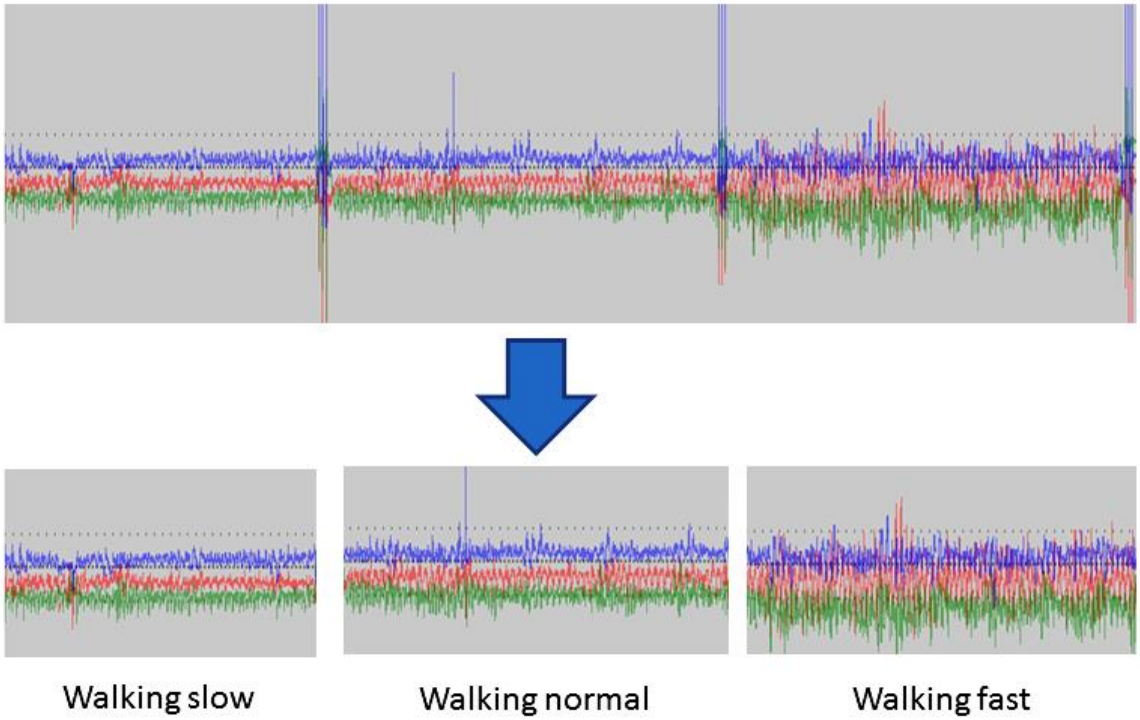

Figure 2-2 Example of Activity Annotation

3 Activity Recognition Algorithm

The SELFBACK activity recognition system uses a supervised machine learning approach. This approach consists of 4 main steps which are: windowing, labelling, feature extraction and classifier training, as illustrated in Figure 3-1.

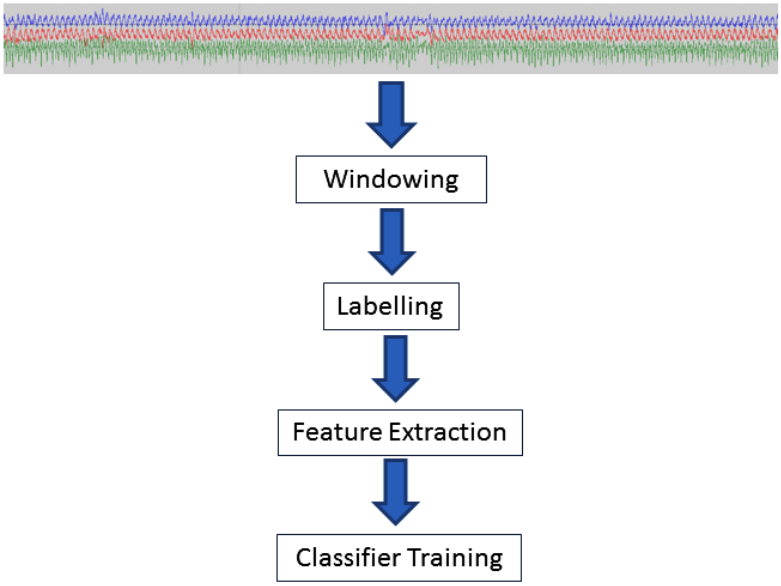

Figure 3-1 Steps of Activity Recognition Algorithm

3.1 Windowing

Windowing is the process of partitioning collected training data into smaller portions of length *l*, specified in seconds. When choosing *l*, our goal is to find the window length that best balances between accuracy and latency. Shorter windows typically produce less accurate activity recognition performance while longer windows produce latency, as several seconds worth of data need to be collected before a prediction is made. In this system, we select a window size of 10 seconds, after analysing a number of different window sizes ranging from 2 second to 30 seconds (See Section 4 for results). Windows are overlapped by 0.5 of their length (5 seconds) along the data stream as illustrated in Figure 3-2. Each extracted window is treated as a separate instance for use in training our classifier.

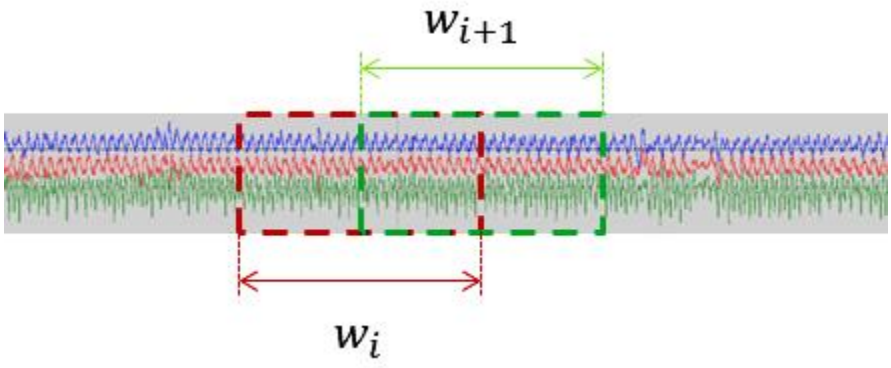

Figure 3-2 Illustration of accelerometer data windowing

3.2 Labelling

Once windows have been extracted, each window needs to be associated with a class label. By default, this is the label of the activity stream from which the window was extracted. Recall from Section 2 that data was collected for nine activities (see Table 1). However, we observed that the more granular the activity labels, the more activity recognition accuracy suffers. In the case of some closely related classes e.g. sitting and lying, it is very difficult to distinguish between these classes from accelerometer data recorded on the wrist. This is because the placement of the wrist is very similar for these activities. Also, for activity classes distinguished by intensity (i.e. walking slow, walking normal and walking fast) the distinction between these activity classes can be more subjective than objective. Because the pace of walking is self-selected, one participant's slow walking pace might better match another's normal walking pace. Both these problems have the effect of artificially limiting the performance of the classifier.

In order to mitigate against these problems, the set of five activity labels in Table 2 are used to train the final activity classifier. Evaluation results for activity recognition for both the 9 classes and the 5 classes are presented in Section 5.

Table 2 Activity labels used for training our activity recognition system and description of the specific activities under each label

Activity Label	Specific Activities
Running	Jogging
Walking	Walking slow, walking normal and walking fast
Standing	Standing
Sedentary	Sitting, Lying
Stairs	Up stairs, down stairs

3.3 Feature Extraction

The next step is to extract features to represent these instances (which are essentially each of the windows). Many different feature extraction approaches have been proposed for accelerometer data for the purpose of activity recognition (Lara & Labrador, 2013). Most of these approaches involve extracting statistics e.g. mean, standard deviation, percentiles etc. on either the raw accelerometer data (time domain features) or on coefficients of discrete Fourier transforms applied to the raw data (frequency domain features). While these approaches have produced good results, we use a novel approach that directly uses coefficients obtained from applying discrete cosine transforms on the raw accelerometer data as features. This approach has outperformed both statistical time-domain and frequency domain features in our comparative evaluation (see Section 4 for results).

The use of discrete cosine transforms (DCT) for extracting features from accelerometer data was proposed in (He & Jin, 2009). In this approach, DCT is applied to each axis (x, y and z) of the accelerometer data. For each axis, a set of DCT coefficients are returned which are an expression of the original accelerometer data in terms of a sum of cosine functions at different frequencies. The main difference between our approach and the approach in (He & Jin, 2009) is that we include the magnitude $m = \{m_1 \dots m_l\}$ of the accelerometer data for each window as a separate axis, where m_i is defined as:

$$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

Thus, our approach uses 4 axes: *x*, *y*, *z* and *m*. Including magnitude in our representation helps to make our system less sensitive to changes in orientation of the sensing device. DCT is applied to each one of these axes to obtain a set of DCT coefficients x', y', $z'$and m'. DCT compresses all of the energy in the original data stream into as few coefficients as possible and returns an ordered sequence of coefficients such that the most significant information is concentrated at the lower indices of the sequence. This means that higher frequency DCT coefficients can be discarded without losing information. On the contrary, this might help to eliminate noise. Thus, in our approach we retain only the first 48 coefficients of x', y', $z'$and m'. This process is illustrated in Figure 3-3.

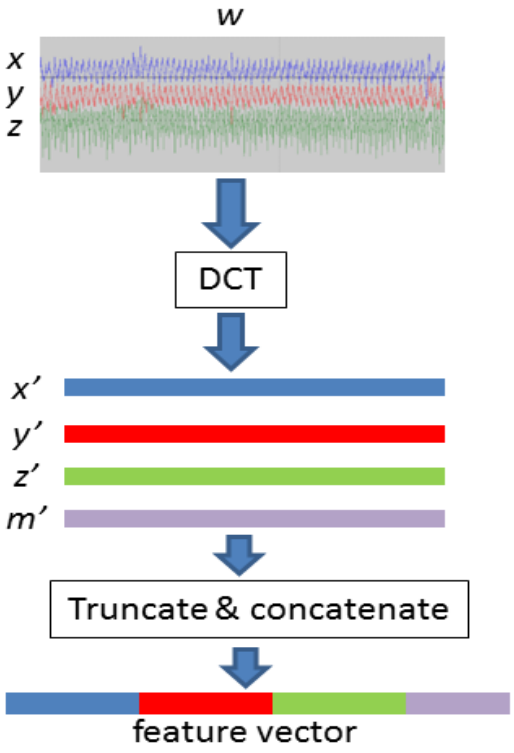

Figure 3-3 Feature extraction using DCT

The coefficients returned after applying DCT are combinations of negative and positive real values. For the purpose of feature representation, we are only interested in the magnitude of the DCT coefficients, irrespective of its (positive or negative) sign. Accordingly for each DCT coefficient e.g. x_i', we take the absolute value $|x_i'|$. The final feature representation is obtained by concatenating the absolute values of the first 48 coefficients of x', y', z' and m' to produce a feature vector of length 192.

3.4 Classifier Training

Once all instances (windows) are represented in the space of 192 DCT coefficients, these representations are then used to train a support vector machine (SVM) classifier. SVM was chosen because of its superior performance in our comparative evaluation, over other classifiers e.g. kNN, logistic regression, decision trees and Naïve Bayes (see Section 4 for evaluation results)

3.5 Step Counting

An important piece of information that can be provided for ambulation activities is a count of the steps taken. This information has a number of valuable uses. Firstly, step counts provide a convenient goal for daily physical activity. Health research has suggested a daily step count

of 10,000 steps for maintaining a desirable level of physical health (Choi, Pak, Choi, & Choi, 2007). A second benefit of step counting is that it provides an inexpensive method for estimating activity intensity. Step rate thresholds have been suggested in health literature that correspond to different activity intensities. For example, (Abel, Hannon, Mullineaux, & Beighle, 2011) identified that step counts of 94 and 125 steps per minute correspond to moderate and vigorous intensity activities respectively for men, and 99 and 135 steps per minute correspond to moderate and vigorous intensity activities for women. Accordingly, step counts might provide a more objective approach for identifying activity intensity in the SelfBACK system than classifying different walking speeds.
Accordingly, we propose two commonly used approaches involving frequency analysis and peak counting algorithms for inferring step counts from accelerometer data.

3.5.1 Frequency Analysis

The main premise of this approach is that frequency analysis of walking data should reveal the heel strike frequency (i.e. the frequency with which the foot strikes the ground when walking) which should give an idea of the number of steps present in the data. (Ahanathapillai, Amor, Goodwin, & James, 2015). For walking data collected from a wrist-worn accelerometer, one or two dominant frequencies can be observed, heel strike frequency, which should always be present, and the arm swing frequency which may sometimes be absent. Converting accelerometer data from the time domain to the frequency domain using Fast Fourier Transform (FFT) should enable the detection of these frequencies. For step counting, this approach seeks to isolate the heel strike frequency. Accordingly, the step count can be computed as a function of the heel strike frequency. For example, for frequency values in Hertz (cycles per second), the step count can be obtained by multiplying the identified heel strike frequency with the duration of the input data stream in seconds.

3.5.2 Peak Counting

The second approach involves counting peaks on low-pass filtered accelerometer data where each peak corresponds to a step. This process is illustrated in Figure 3-4. For filtering, we use a Butterworth low-pass filter with a frequency threshold of 2 Hz for walking and 3 Hz for running. The magnitude *m* of the accelerometer signal is obtained by combining the *x*, *y* and *z* axes. The low-pass filter is then applied on m to obtained a filtered signal m', which has the benefit of filtering all frequencies in m that are outside of the range for walking and running respectively. In this way, any changes in acceleration left in m can be attributed to the effect of walking or running. A peak counting algorithm is then deployed to count the peaks in m where the number of peaks directly corresponds to the count of steps.

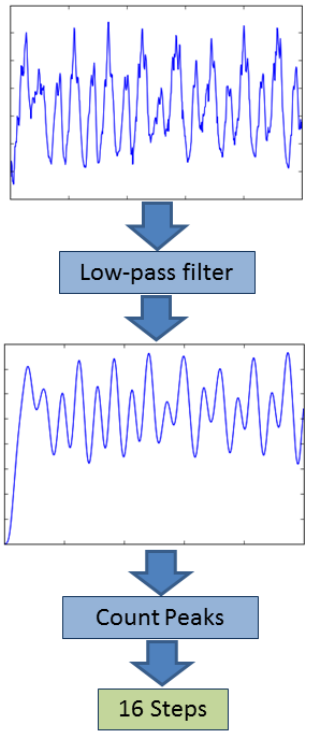

Figure 3-4 Peak Counting Method for Step Counting

4 Evaluation

In this section we present evaluation of the decisions that have guided the development of our system. These decisions include choice of window size, classifier and feature representation. We also present evaluation results for activity classification using all 9 activity classes (see Section 2) and using the 5 chosen activity classes (see Section 3.2). Finally, we present results from two step counting algorithms.

Our experiments are reported using a dataset of 20 users. Evaluations are conducted using a leave-one-person-out strategy i.e. one user is used for testing and the remaining 19 are used for training. Performance is reported using macro-averaged F1.

4.1 Window Size

Our first evaluation addresses the question of window size. For these experiments, SVM classifier and DCT feature representation is used. While we address the questions of classifiers and representation in subsequent paragraphs of this section, it is difficult to present results for every combination of widow size, feature representation, classifier and number of classes. Thus, for the sake of brevity, we keep all parameters constant and only vary the parameter we are evaluating. For the parameters we keep constant, we choose settings that have produced the best results. For example, in Table 3, we present F1 scores for varying window sizes ranging from 2 to 60 seconds, while using DCT for feature representation, SVM for classification and using 5 class labels.

Table 3 Activity Recognition Performance at different window sizes

Window Size	F1 Score	Sensitivity	Specificity
2	0.878	0.88	0.967
3	0.885	0.89	0.969
4	0.891	0.89	0.971
5	0.890	0.89	0.971
10	0.906	0.91	0.975
20	0.849	0.85	0.960
30	0.778	0.79	0.943
40	0.720	0.75	0.929
50	0.657	0.69	0.914
60	0.651	0.70	0.881

Note from Table 3 that the highest F1 score is achieved using a window size of 10. Accordingly, all subsequent evaluations in this Section and the final SELFBACK activity recognition system use a window size of 10 seconds.

4.2 Classifiers

Next, we present comparative results for 5 different classification algorithms: SVM, nearest neighbour, decision tree and naïve Bayes classifiers. Here also, DCT feature representation strategy and 5 class labels are used. The F1 scores for these classifiers are shown in Table 4.

Table 4 Activity Recognition Performance of Different Classifiers

Classifier	F1 Score	Sensitivity	Specificity
SVM	0.906	0.907	0.975
Nearest Neighbour	0.857	0.865	0.962
Decision Tree	0.780	0.781	0.942
Naïve Bayes	0.661	0.685	0.922
Logistic Regression	0.877	0.877	0.967

Observe that the best F1 score is achieved using SVM.

4.3 Feature Representation

In this section we present a comparison of different feature representation approaches: DCT, statistical time domain and frequency domain. Time domain features are adopted from (Zheng, Wong, Guan, & Trost, 2010) and are shown in Table 5. Each statistical feature is computed for each one of four axes, *x*, *y* and *z*, and the magnitude *m* from a time window *w*.

Table 5 Time Domain Features

Feature
Sum
Mean
Standard Deviation
Coefficients of variation
Peak-to-peak amplitude
Interquartile range
lag-one-autocorrelation
Skewness
Kurtosis
Signal Power
Log-energy
Zero crossings

The frequency domain features used are adopted from (Tessem & Hessen, 2015). Computing frequency domain features begins with transforming the *x*, *y* and *z* axes of each time window *w*, from the time domain to the frequency domain using Fast Fourier Transforms (FFT) to obtain a set of FFT coefficients f_x, $f_y$and f_z respectively. The statistical measures in Table 6 are then applied to each set of FFT coefficients to obtain the frequency domain feature

representation of w. Accordingly, w is represented in the feature space of the 7 features in Table 6.

Table 6 Frequency Domain Features

Feature
Mean
Standard Deviation
Maximum
Median
Spectral Centroid
Dominant Frequency
Spectral Entropy

Results of our comparative analysis are presented in Table 7. As can be observed, the best performance is achieved using DCT features.

Table 7 Activity Recognition Performance of Different Representations

Representation	F1 Score	Sensitivity	Specificity
Time domain	0.827	0.848	0.958
DCT	0.906	0.907	0.975
Frequency domain	0.813	0.821	0.950

4.4 Classification Granularity

Recall from Section 3.2 that data was collected for 9 different activities. However, only 5 different activity labels are used to train the classifier. Here we compare activity recognition performance with 5 classes, to the performance when all 9 classes are utilised. Results are presented in Table 8.

Table 8 Activity Recognition Performance at Different Levels of Granularity

Number of classes	F1 Score	Sensitivity	Specificity
9 Classes	0.6880	0.691	0.961
5 classes	0.906	0.907	0.975

Observe that better classification performance is achieved for activity recognition using 5 classes. A more detailed analysis is required to better explain the difference in F1 score between activity recognition using 5 classes compared to 9 classes. Table 9 shows precision, recall and F1 measures per class for both 9-class and 5-class classification scenarios. Confusion matrices for the two scenarios are also provided in Table 10 and Table 11 respectively, where the columns represent the predicted classes and the rows represent the actual class labels.

Table 9 Precision, Recall and F1 Scores Per Class

Activity Class	Precision	Recall	F1 Score
9 Activity Classes			
lying	0.56	0.46	0.50
sitting	0.55	0.64	0.59
standing	0.89	0.89	0.89
downstairs	0.86	0.62	0.72
upstairs	0.56	0.58	0.57
jogging	0.86	0.99	0.92
walking fast	0.72	0.65	0.68
walking normal	0.58	0.57	0.57
walking slow	0.65	0.72	0.68
5 Activity Classes			
sedentary	0.96	0.97	0.96
standing	0.94	0.86	0.90
running	0.88	0.99	0.93
stairs	0.87	0.74	0.80
walking	0.89	0.92	0.90

Table 10 Confusion Matrix of 9 Class Activity Classification

		Predicted								
		lying	sitting	standing	jogging	upstairs	down stairs	walk fast	walk normal	walk slow
Actual	lying	115	127	8	0	0	0	0	0	1
	sitting	84	161	4	2	1	0	0	0	0
	standing	6	6	212	2	2	0	0	0	11
	jogging	0	0	0	284	1	0	0	1	0
	upstairs	0	0	3	8	92	7	7	11	30
	down stairs	1	0	1	5	31	89	5	4	8
	walk f	0	0	1	21	3	1	157	53	6
	walk n	0	0	1	4	11	3	48	141	41
	walk s	0	1	7	3	23	4	0	32	181

Table 11 Confusion Matrix of 5 Class Activity Classification

		Predicted				
		sedentary	standing	running	stairs	walking
Actual	sedentary	490	7	2	1	3
	standing	17	205	2	1	14
	running	0	0	283	0	3
	stairs	3	0	9	223	67
	walking	3	5	24	31	679

Observe from Table 9 that the classes with the lowest F1 score, out of the 9 classes in the evaluation, are lying, walking normal and upstairs. The reason for the score can be observed

in Table 10 where, for the activity class lying, only 115 instances are correctly classified and 125 instances are incorrectly classified as sitting. Similarly, 84 instances of sitting are incorrectly classified as lying. This indicates a high degree of resemblance between lying and sitting which can be attributed to the fact that the wrist placement for these two activities is very similar. Hence, differentiating between these two activities is very difficult using accelerometer data collected from the wrist. However, both sitting and lying represent sedentary behaviour and it makes sense for both activities to be combined into a class called Sedentary.

A similar explanation can be observed for walking normal were 48 instances are incorrectly classified as walking fast and 41 as walking slow. Accelerometer data for walking at different speeds will naturally be very similar. Also, the same walking speed is likely to be different between participants due to the subjectivity inherent in users' judgment of their walking speeds. In addition, a user may unconsciously vary their pace while trying to adhere to a specific walking speed. Again these reasons make it more useful to have the three walking speeds combined into one class called Walking and have walking pace computed as a function of step rate.

In Table 10, the class upstairs appears to be most similar to walking slow but also has similarities with walking normal, walking fast and jogging. A less granular set of class labels is likely to lead to more accurate classification.

Note that for the 5 class classification scenario in Table 9, 4 of the 5 classes have F1 scores greater than 0.9. This is a significant improvement from 9 class scenario where only jogging has an F1 score of over 0.9. The Stairs activity class (F1 = 0.80) is the only one with an F1 score below 0.9. Note that this relatively low F1 score is largely due to the low precision (0.74) achieved for this activity class. Observe from the confusion matrix in Table 11 that 67 instance of stairs are incorrectly classified as walking. This highlights the difficulty in separating between walking on flat surface, and walking up or down stairs. This is expected as both activities are very similar except for the nature of the surface they are performed on.

4.5 Step Counting

This final sub-section presents an evaluation of our step counting algorithms. For this, we collected a separate set of walking and running data with known actual step counts. This was necessary because actual counts of steps were not recorded for the initial dataset collected. In total, 19 data instances were collected for walking and 11 for running. For walking, participants were asked to walk up and down a corridor while counting the number of steps they took from start to finish. Example script is shown in Figure 4-1. Reported step counts for walking range from 244 to 293. Participants performed a number of different hand poses which included walking with normal hand movement, with hands in trouser pocket and carrying a book for coffee mug. Walking data also included one instance of walking down as set of stairs (82 steps) and one instance of walking up a set of stairs (78 steps).

Protocol – Step Counting

1. Put on wristband
2. Clap 3 x
3. Walk to the end of the corridor and back at self-selected walking speed while counting your steps
4. Clap 3 x
5. remove wristband

Figure 4-1 Example Script for Step Counting

Running data was collected on a treadmill. Participants were requested to run a treadmill at a self-selected speed for a self-selected duration of time. Here also, three claps were used to mark the start and end of the running session. Two participants standing on the side were asked to count the steps in addition to the runner, due to the difficulty that may be involved in running and counting steps at the same time. Reported step counts for running range from 150 to 210.

The objective of this evaluation is to match, for each data instance, the count of steps predicted by each algorithm, to the actual step counts recorded. Root means squared error (RMSE) is used to measure performance. Because both step counting algorithms do not require any training, all 30 data instances are used for testing. Evaluation results are presented in Table 12

Table 12 Performance of Step Counting Algorithms Measured using Root Mean Squared Error

Step Counting Algorithm	RMSE Walking	RMSE Running
Frequency Analysis	11.245	6.250
Peak Counting	6.374	5.576

Note that better performance is observed from the Peak Counting method, thus this has been set as the default step counting approach for the SELFBACK system.

5 Conclusion

In this document, we have presented the SELFBACK activity recognition system. The system accepts tri-axial accelerometer data as input and uses a supervised machine learning classifier for automatically inferring activity from the input. Our approach for collecting data has been discussed. We have presented the activity recognition algorithm and its stages i.e. windowing, feature extraction, labelling and classifier training. We discussed the parameter choices for the system at each one of these stages. We also presented two approaches for inferring step counts from the accelerometer data. Finally, we presented a detailed evaluation of the system, along with evaluations to justify our choices of parameters.

6 References

Abel, M., Hannon, J., Mullineaux, D., & Beighle, A. (2011). Determination of step rate thresholds corresponding to physical activity intensity classifications in adults. *Journal of Physical Activity & Health*, *8*, 45–51.

Ahanathapillai, V., Amor, J. D., Goodwin, Z., & James, C. J. (2015). Preliminary study on activity monitoring using an android smart-watch. *Healthcare Technology Letters*, *2*(1), 34–9. doi:10.1049/htl.2014.0091

Choi, B. C. K., Pak, A. W. P., Choi, J. C. L., & Choi, E. C. L. (2007). Daily step goal of 10,000 steps: A literature review. *Clinical and Investigative Medicine*, *30*(3), 146–151.

He, Z., & Jin, L. (2009). Activity recognition from acceleration data based on discrete consine transform and SVM. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, (October), 5041–5044. doi:10.1109/ICSMC.2009.5346042

Lara, D., & Labrador, M. A. (2013). A Survey on Human Activity Recognition using Wearable Sensors. *IEE Communications Surveys & Tutorials*, *15*(3).

Tessem, A. J., & Hessen, H. (2015). Human Activity Recognition with two Body-Worn Accelerometer Sensors, Internal Report NTNU, (December 2015).

Zheng, Y., Wong, W., Guan, X., & Trost, S. (2010). Physical Activity Recognition from Accelerometer Data Using a Multi-Scale Ensemble Method. *Proceedings of the Twenty-Fifth Innovative Applications of Artificial Intelligence Conference*, 1575–1581.