Vietnam National Universities – HCMC
International University
Biomedical Engineering Department

HUMAN ACTIVITY RECOGNITION USING SMARTPHONES

by

VU NGOC THANH SANG

A thesis submitted to the Biomedical Engineering Department in partial fulfillment of the requirements for the degree of Engineer

Ho Chi Minh City, Vietnam

July, 2014

Vietnam National Universities – HCMC
International University
Biomedical Engineering Department

HUMAN ACTIVITY RECOGNITION

USING SMARTPHONES

APPROVED BY: Advisor

____

Dr. Nguyen Duc Thang

APPROVED BY: Committee

____

Prof. Vo Van Toi - Chair

____

Dr. Nguyen Thanh Hai

____

Dr. Pham Thi Thu Hien

ACKNOWLEDGMENTS

It is with deep gratitude and appreciation that I acknowledge the professional guidance of Dr. Nguyen Duc Thang. His constant encouragement and support helped me to achieve my goal.

Besides my advisor, I would like to thank Dr. Truong Quang Dang Khoa whose encouragement, guidance and support me to develop an understanding of one method in my research.

In addition, this thesis could not be improved and corrected without the valuable recommendations of my committee Dr. Nguyen Thanh Hai.

Last but not the least, I would like to thank my peers for sharing their experiences and giving me advices to support me spiritually throughout my research.

Contents

List of tables

List of figure

ABSTRACT

The sedentary lifestyle is becoming popular especially for intellectual works. Although a physical inactivity lifestyle may cause many unexpected illnesses, it is complicated to build up a positive lifestyle due to the lack of reminder systems to manage and monitor physical activities of people. This research represents an effective way for daily activity monitoring using accelerometer and gyroscope sensors embedded in a smartphone. Signals were recorded from accelerometer and gyroscope sensors while a user was wearing the smartphone and performing different activities (going downstairs, going upstairs, sitting with the phone in a pocket, driving and putting the phone on the table). For offline analysis, the classification algorithms with k-nearest-neighbor (kNN), artificial neural network (ANN) and support vector machine (SVM) were applied to recognize user's activities. The overall accuracy of recognizing five activities was 74% for kNN, 75.3% for ANN and 94.5% for SVM respectively. The kNN and SVM classification were later on implemented on the smartphone for analysis and comparison. Based on the recognized activities during a day, users are able to manage their daily activities for a better life.

Keywords — *Smartphone, activity recognition, activity monitoring.*

CHAPTER I
INTRODUCTION

1.1. Problem statement

As reported by the World Health Organization, 3.2 million annual deaths were associated with physical inactivity lifestyle [1]. In 2004, it is the fourth risk factor causes of death in the world. People who have physical inactivity lifestyle and high body mass index (BMI) will have a high risk of high blood pressure. The more inactive people are, the more risky they have diseases.

	Risk factor	Deaths (millions)	Percentage of total
	World		
1	High blood pressure	7.5	12.8
2	Tobacco use	5.1	8.7
3	High blood glucose	3.4	5.8
4	Physical inactivity	3.2	5.5
5	Overweight and obesity	2.8	4.8
6	High cholesterol	2.6	4.5
7	Unsafe sex	2.4	4.0
8	Alcohol use	2.3	3.8
9	Childhood underweight	2.2	3.8
10	Indoor smoke from solid fuels	2.0	3.3

Figure 1: Ranking of selected risk factors that cause of death, 2004

In order to have a better health as well as prevent cardiac diseases, people at all ages should practice physical activities at least thirty minutes per day according to United States Department of Health and Human Services [2]. These activities may include gardening, bicycling, fast dancing, two-mile walking, stairs walking, and playing sports, etc. Such regular physical activities can decrease premature death by preventing heart diseases, reducing diabetes risks and controlling high blood pressure. However, despite the great benefits that physical activities offer, it is complicated to build up a good habit to achieve the desired healthy lifestyle. One of the major obstacles is that most people are too busy to spend at least thirty minutes per day to perform physical exercises. The best they can do is to practice any activities whenever it is possible, leading to a problem that they cannot manage duration as well as intensity of those physical activities.

1.2 Objectives and scope of project

The scope of thesis is developing a smartphone application that can recognize human activities and calculate the energy expenditure. In details, the application receives six inputs from accelerometer and gyroscope sensors then generates two outputs as a desired activity and consumed energy.

The objective of this thesis is to apply the smartphone application in healthcare. Particularly, according to the activities and energy expenditure of users, it is possible to monitor their lifestyle in order to improve the quality of their lives. Furthermore, users can manage their works and exercises effectively.

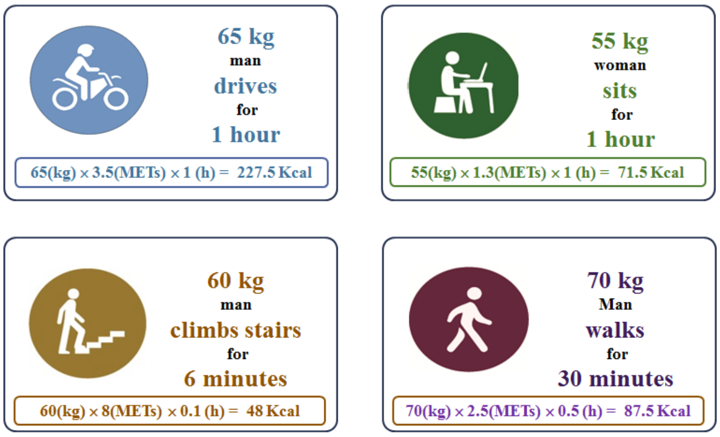


Figure 2: Energy measurement from daily physical activities

1.3. Limitations

There are several limitations in my thesis. Firstly, many classification methods such as k nearest neighbors, artificial neural networks, support vector machine have been applied, while they are not yet optimally themselves. Secondly, there is one subject in my experiment. Therefore, the accuracy of activity recognition may vary when the activity recognition is applied on other people. Thirdly, the activities were recorded separately and independently. As the result of this, the subject needs to perform activities for a while in order to get right recognitions. Additionally, when the subject changes from one activity to another, the recognition will be confused for a short time. These limitations will be eliminated in the future work.

1.4 Thesis organization

I organized the chapters in my thesis so that the material and information within each chapter is organized to establish the conceptual foundation on which my thesis has been developed. The rest of my thesis is divided into 6 relatively short chapters, each with a sufficient and manageable amount of information. The chapters are divided into the following 6 parts:

.

- *Chapter 2: Literature review*. This chapter describes related works that have been done by many other researchers. Doing this literature review provides me an overview of the current research topics as well as a clear idea of what has already been done in the field. I am able to see the methods that other experienced researchers have used and then decide my methodological approach.

- *Chapter 3: Methodology*. This chapter discusses the way I have done my research. In more detail, in this chapter, I will introduce about the smartphone, the way I processing data, how I extract features of signals and classify them.
- *Chapter 4: Smartphone implementation*. This chapter shows how I implemented my idea on a Blackberry 10 smartphone. It also gives the overview of the programming language that the smartphone support. In addition, the information of interaction between QML and C++ as well as data accession are provided.
- *Chapter 5: Result*. This chapter contains the accuracy of my method in both offline and online tasks. The offline task was done on a computer, while the online task was done on the smartphone.
- *Chapter 6: Conclusion and discussion*. This chapter is the evaluation of my thesis. In particular, it discusses about advantages and disadvantage of my work. The final part of this chapter is my future work in this field.

CHAPTER 2
LITERATURE REVIEW

Different approaches have been proposed to monitor users' activities by using accelerometer and gyroscope sensors. They can be divided into two major approaches: wearable sensor based-approaches and smartphone-based approaches.

2.1 Wearable sensor-based approaches

In the wearable sensor-based approaches, three gyroscope and accelerometer sensors were used to measure joint angles including hip, knee and ankle joints of monitoring subjects for walking, sit-to-stand, and stand-to-sit activities [3]. In result, coefficient of correlation ranged from 0.847 to 0.986 and root mean square error was from 2.8 to 4.73 for different activities.

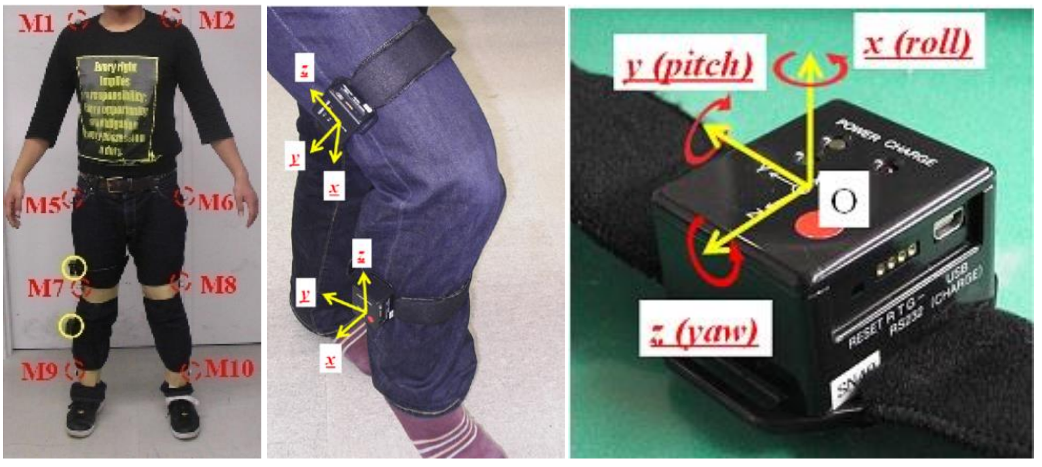


Figure 3: The locations of gyroscope and accelerometer sensors of the experiment in [3].

Another sensor-based approach used only accelerometer located on participants' hand wrists, waists, and ankles [4]. The proposed activities consist of static, home activities, upstairs, downstairs, walking, running, bicycling. The dimension of features was narrowed down by linear discriminate analysis (LDA) and sent to a knearest neighbor (k-NN) classifier for classification. The overall accuracy ranged from 85.16 to 96.98 percent.

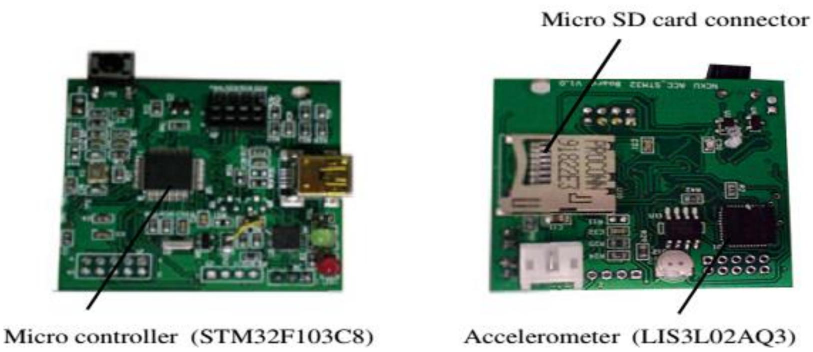


Figure 4: The wearable physical activity sensor module

Activity recognition researches based on wearable sensors required combinations of sensor. One research was that each of eight subjects worm five sensor units on their chest, arms, legs for recognition of nineteen daily and sport activities [5].

Figure 5: Sensor modules and their positioning on the body

Each sensor unit is included a gyroscope, an accelerometer and a magnetometer. The classification techniques used in this study consisted of Bayesian decision making (BDM), the least-squares method (LSM), the k-nearest neighbor (kNN) algorithm, dynamic time warping, support vector machines, and artificial neural networks. The average correct classifications of BDM, LSM, kNN, SVM, ANN were 99.15, 89.5, 98.45, 98.7 and 91.55 respectively.

Another research combined an accelerometer and a capacitive proximity sensor in order to detect nine basic activities of with seven participants [6]. The author used support vector machine as a classification method due to its high relevance and fast performance in activity recognition. The F-measure of classification method ranged from 67.2 to 73.5% for different activities.

Figure 6: The inertial data logger includes a 3 axis accelerometer, a microSD flash card for storing the sensor data and a USB connector for accessing the data

Furthermore, an integrated ear-worn activity recognition (e-AR) sensor has been developed in [7]. Accelerometer values from the e-AR were combined with vision sensor by using a Gaussian Mixture Model Bayes classifier. The proposed activities were walking, standing, standing (head tilted), sitting, sitting (sofa), reading, eating, lounging and lying down. The result of e-AE combined ambient sensing (82.78) is significant increasing in average accuracy in comparison to the implemented of the e-AR sensor alone (72.78 %).

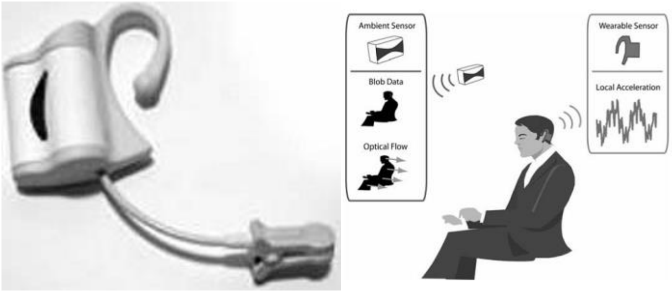


Figure 7: The e-AR sensor (left) and the schematic diagram of the proposed ambient and wearable monitoring system (right).

In another work, researchers developed a novel wearable system that can be used easily and portable. Participants were three woman and eleven men with age between 27 and 35. Then they were asked to perform several activities such as climbing stairs, walking, talking, standing, working. The F-Measure obtained for each class is obtained using random forest. In particular, activity with the best performances are walking (95.9 %) and working at computer (97.7 %). In addition, the F-measure of talking, climbing stairs and standing were 92.9, 89.8, 88.8 percentage respectively.

Figure 8: The components of the wearable system (left) and the wearable system worn by an experimenter (right)

2.2 Sensor embedded in smartphone approaches

Alternatively, with the widespread of smartphones nowadays, many researchers have currently focused on existing sensors embedded in a smartphone for the task of recognizing users' activities, e.g. the implementation of real time activity recognizer [8]. The achieved accuracies ranged from 70% to 90% for different activities consisting of walking, climbing downstairs, climbing upstairs, sitting down, standing up, falling and so forth.

Figure 9: Sceenshot of smartphone in [8]

In another aspect, a smartphone was used to detect falls in human [9]. Subjects were asked to repeat both falls as well as normal actions such as walking, sitting down and jumping to generate training dataset. Observing the differences of the two activities, researchers investigated adaptive thresholds to distinguish them in a feature space.

Figure 10: The implementation of E-FallD

Utilizing a smart phone as a reminder to ask people to take regular breaks from prolonged sitting seems promising since it motivated a mobile-phone user to become more active. In an application described in [10], a smartphone generated acoustic warnings and vibration with virtual persuasive messages to remind a user to take a break. In another research, three activities such as sitting, standing and walking were classified by a single tri-axial accelerometer embedded in smartphone. The researchers used Naive Bayes, Bayes Net algorithms to classify those activities and achieved the accuracy from 84.21 % to 98.86 % for different period of times.

Figure 11: SitCoach main screen

Recognizing the physical activities in the natural setting where the mobile phone's position and orientation are varying, due to the position, material and size of the users' pocket. Developing in cases of 6 pocket positions, this research applied SVM classifier to recognize 7 common physical activities. The overall F-score increases to 94.8% due to the determined pocket position.

Figure 12: Pocket locations and phone orientations

CHAPTER 3
METHODOLOGY

Figure 13: The process of activity recognition

The overall system was described in Fig. 1 where the signals from an accelerometer and a gyroscope of a mobile phone were preprocessed and inputted into Feature extraction module to extract main features. Certain characteristics of signals such as auto regressive (AR) coefficients, fractal dimension, mean and standard deviation were calculated. The features were later classified with k-nearest neighbor (kNN), artificial intelligence network (ANN) and support vector machine (SVM) to recognize different activities in offline classification. In online classification, I applied both kNN and SVM method in activity recognition. According to the activities and the duration of them, the energy expenditure is estimated.

3.1. Device

3.1.1*) Hardware*

The system was implemented on the infrastructure of Blackberry Z10, STL100-2 with a dual-core central processing unit (CPU) running at 1.5 GHz, 2 GB RAM and 16GB storage. The phone is 130 x 65.6 x 9 mm in size and 137.5 grams in weight. The touch screen has a 768 x 1280 resolution. Most importantly, the phone supports an accelerometer, gyroscope, proximity and compass sensors. However, this paper used only accelerometer and gyroscope sensors to record motions from users. Electronic accelerometer and gyroscope sensors have influenced effectively physical activity detection in living subjects since the accelerometer respond to changing velocities of subject's movements and the gyroscope sensor responded to changing orientation of a subject's movements.

3.1.2*) Software*

Our application was developed on the Blackberry 10 Native Platform. Blackberry 10 Native is an open source framework designed for mobile devices with a powerful Software Development Kit (SDK) based on the combination of C++ and Qt/QML framework [11]. Blackberry 10 Native SDK provides necessary drivers needed to interact with the hardware and libraries needed to deploy a smartphone application. Furthermore, Blackberry 10 Native allows programmers to access accelerometer and gyroscope sensors and to save their values into a log for further analyses.

3.2. Collecting data

Our approach required two recorded datasets for training and testing the classifier. The activity labels of both training and testing were stored in a database, hence the accuracy of our approach was evaluated. I recorded accelerometer and gyroscope sensors' values and the corresponding time index. The accelerometer was designed to detect changes due to movements and vibrations of the phone. The gyroscope sensor was used to measure orientation and rotation of the phone.

The subject for my experiment was a male, 22 years old. His height and weight were 1.7 m and 75 kg. The smartphone is located at the right jean's pocket of the subject.

For offline classification, there were five labeled activities including going downstairs, going upstairs, driving, sitting, and putting the phone on the table. They were divided into two set: training set and testing set. The training set had 120 data, while the testing set had 80 data. In details, the training set included 15 data of walking downstairs, 30 data of driving, 30 data for sitting, 30 data for putting the phone on table, 15 data for walking upstairs. The testing set consisted of 10 data of walking downstairs, 20 data of driving, 20 data for sitting, 20 data of putting the phone on table, 10 data for walking upstairs. Each data represented by 3 variables of accelerometer sensor and 3 variables of gyroscope sensor for 60 seconds. The features of activities later on were extracted in a computer. The kNN, ANN, SVM classification methods were applied those features.

In smartphone implementation, the information was recording in 10s and updated every second. The purpose of online classification was to recognize four activities such as driving, sitting, climbing stairs and walking. In online classification, the training data was 4500 and the testing data was 500 for each activity. Each data was a feature vector of a activity. The smartphone was implemented to classify information by kNN and SVM method automatically.

Finally, base on recognized activity, the energy the users spent during a period of time was estimated. Different activities make users burn different amount of calories. Therefore, the users are able to know how much energy they have consumed after the period of running the application.

3.3. Feature extraction

3.3.1) *Autoregressive coefficient (AR)*

The most common model used to approach spectrum estimation of random signals is AR model[12]. This model has been attractive in many applications because of its intuitive form, which is the value of a present variable depends on the values in the past. The paper implemented an AR model with periodic parameters. In AR(p) model, the data output is assumed to be generated by its previous values by the following input – output equation:

$$y_n = \sum_{k=1}^{p} a_k y_{n-k} + \varepsilon_n$$

where y_n is the sum of the relation constant of y_{n-k} as the previous values, a_k is the coefficients at the lag k, ε_n is a zero mean white noise process with unknown variance and p is the order of the system.

The main properties of AR(p) models mentioned refer to expected value, variance, autocorrelation, partial autocorrelation. The expected values and variance of parameters can be estimated from historical data. The autocorrelation function satisfies a differential equation known as the Yule-Walker equation. Yule-Walker introduced a method to estimate the parameters of the AR model by the method of moments. The partial autocorrelation function is another representation of a series in term of a time dependent structure.

The Yule-Walker Equations for the general AR(p) model is calculated as following.

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{p-1} \\ y_p \end{pmatrix} = \begin{pmatrix} y_0 & y_{-1} & y_{-2} & \cdots y_{-p} \\ y_1 & y_0 & y_{-1} & \cdots y_{1-p} \\ \vdots & \vdots & \vdots & \vdots \\ y_{p-2} & y_{p-3} & y_{p-4} & \cdots y_{-1} \\ y_{p-1} & y_{p-2} & y_{p-3} & \cdots y_0 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{p-1} \\ \phi_p \end{pmatrix}$$

Substituting y0 = 1 in the equation ( ) gives

$$\underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{p-1} \\ y_p \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 1 & y_{-1} & y_{-2} & \cdots y_{-p} \\ y_1 & 1 & y_{-1} & \cdots y_{1-p} \\ \vdots & \vdots & \vdots & \vdots \\ y_{p-2} & y_{p-3} & y_{p-4} & \cdots y_{-1} \\ y_{p-1} & y_{p-2} & y_{p-3} & \cdots 1 \end{pmatrix}}_{R} \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{p-1} \\ a_p \end{pmatrix}}_{\Phi}$$

Besides, we can write the above equation in the matrix multiplication form

$$Y = R\Phi$$

In order to estimate an AR coefficient, which is the column matrixΦ, I perform the following calculation:

$$\Phi = R^{-1}Y$$

The AR(p) coefficients were estimated from each parameter and concatenated together to form the feature vectors for classification. The number of AR(p) coefficients for a parameter in this work was 10. Therefore, the concatenated vectors of each activity have 60 AR coefficients.

3.3.2*) Fractal dimension*

Fractal dimension presents the irregular characteristic of a time series based on the power law index [13]. Although the fractal dimension can be considered similar to Euclidian dimension, there were significant differences between them. The main difference lied in the fact that fractal dimension may be a non-integer. The Euclidian dimension represents a mathematical system of a set, while the fractal dimension demonstrates how a set in micro-scale is similar to its counterpart in the macro-scale.

The closest integer value of a fractal dimension shows its equivalence to the Euclidian dimension. For example, if the fractal dimension is close to zero, it describes a point (0-dimension in Euclidian). It describes lines when it approaches one (1-dimension in Euclidian) and surfaces when it approaches to two (2-dimension in Euclidian), and so on.

Higuchi's algorithm is an effective method to calculate the fractal dimension in time domain [14]. It is based on the calculation of the length of the curve in time series. The length $L_m(k)$ of the curve has k-segments is calculated by the following formula:

$$L_m(k) = \frac{1}{k}\left\{\left(\sum_{i=1}^{\left[\frac{N-m}{k}\right]} \left|X(m+ik) - X(m+(i-1)k\right|\right)\frac{N-1}{\left|\frac{N-m}{k}\right|k}\right\}$$

where N is the number of data samples, m is an integer smaller than k $(m = 1,2,\ldots,k)$. The relationship of the length $L_m(k)$ and the fractal dimension D is shown as the following formula:

$$\langle L(k)\rangle \sim k^{-D}$$

where $\langle L(k)\rangle$ is the average value over k set of $L_m(k)$.

3.3.3)*Mean*

The use of mean is usually for measuring central tendency of a set, which has only one mean. Without weights assigned on data, mean is referred to the arithmetic mean. The arithmetic mean $\bar{x}$ is equal to the sum of all variables x_n divided by the number of variables N and calculated as follow:

$$\bar{x} = \frac{1}{N}\sum_{n=0}^{N} x_n$$

The mean and standard deviation are the common parameters used in locating central and spread of a set. The main reason for this is that the normal distribution is calculated in terms of these two parameters. This distribution plays an important role in many biological and medical applications.

3.3.4*) Standard deviation*

Although two sets can have the same mean, they are different in case of variability. The standard deviation is the measurement of variability evaluating a range of the data set and the relationship of the mean with the rest of the data. The standard deviation will be small if the data points are close to the mean. On the other hand, if many data points are far from the mean, it reveals that the variance of responses is wide and the standard deviation is large. If data values are equal at all points, then the standard deviation is zero. In addition, the standard deviation is used to predict the power of the signal and calculated using the following formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2}$$

where $\bar{x}$ is the mean of data and N is the number of data points.

3.4. Classification

In classification, I tested my idea firstly with offline classification. The training data is 120 feature vectors with classified labels, while the testing data is 80 feature vectors with unknown vectors. Each vectors included 60 AR coefficients, 6 fractal dimension, 6 means and 6 standard deviations. On the other hand, the offline data consisted of 18000 training data and 2000 testing data. The feature vectors were reduced into 6 fractal dimension, 6 means and 6 standard deviations.

3.4.1*) k-Nearest Neighbor (kNN)*

Firstly, the non-parametric kNN algorithm is used for classification. It is a type of instance-based learning and is the simplest of all machine learning algorithms. The training examples are multidimensional labeled vectors in feature space.

In this research, the neighbors were taken from five classes (going downstairs, going upstairs, sitting, driving, walking and putting the phone on the table). Recorded data with class labels was divided into two sets: training set (classified) and testing set (unclassified). For the recognition of the testing set, the Euclidian distance [15] was computed. Here, the Euclidean distance is the distance between points $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ in a feature space and calculated as follows:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

In the next step, the distance from each sample in the testing set to all the samples in training set is calculated and the minimum value is chosen. Therefore, the classified label is based on the nearest distance of each testing point to the corresponding class in the training set.

3.4.2*) Artificial Neuron Network (ANN)*

a) Introduction

ANN is defined as a computational model based on biological neural networks [16]. Figure 8 shows the model of a neuron which forms an ANN. There are three basic elements of a neural model: weight, summation and activation function. An input signal x_j connected to neuron k is multiplied with its own weight w_{kj}. The summation module adds all the input signals together and combines the result to the bias b_k. The activation potential v_k is the summation of the inputs with weights and bias. The activation function is the function that limits the amplitude of an output of a neuron. Common activation functions are threshold, piecewise and sigmoid functions.

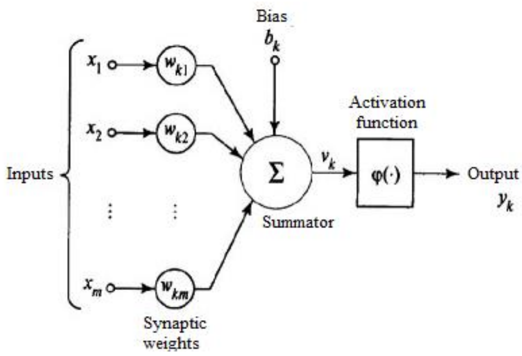


Figure 14: A neural model

An ANN has three layers described as input, hidden and output layers. The input layer receives signals from external environments or other networks. The hidden layer intervenes inputs and outputs in several useful manners. The more added hidden layers, the higher order statistic the network is enabled to extract. The output layer constitutes an overall response of the network to the activation pattern.

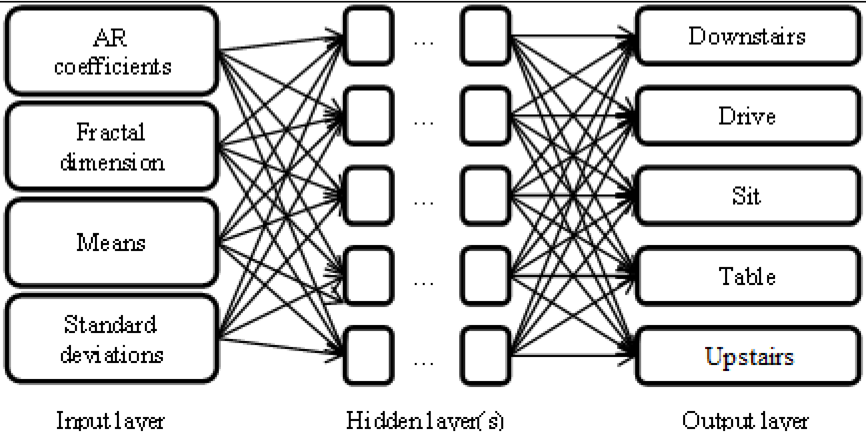


Figure 15: A artificial neural layers

In the experiment, there was feature vectors extracted from 6 raw signals from the two embedded sensors in smartphone. There were 78 inputs into 20 hidden layers to generate a desired output. The output was one of five proposed activities such as walking downstairs, driving motorbike, sitting, putting the phone on a table, walking upstairs. The classification was implemented 10 times randomly to find the average accuracy.

b) Benefits of ANNs

It is can be seen that an ANN has a parallel distributed structure and an ability to learn and generalize information. Generalization is the process of producing reasonable outputs from given inputs during leaning. Several advantages of ANNs [17] are described as the following:

1) **Nonlinearity**: Although an artificial neuron is either linear or nonlinear, a neural network is nonlinear system. Its nonlinearity is very important if the inputs signals are generated by physical mechanism.

2) **Input – output mapping**: The common learning in ANNs is supervised learning in which a set of labeled training samples is modified. Each sample

includes a unique input signal and its corresponding output signal. The network is trained many times until it reaches a steady stage where the synaptic weights have no significant changes. This leaning builds up an input – output mapping for the given problem.

3) **Adaptivity**: Synaptic weights of ANNs have capability to adapt changes of the surrounding environment. In particular, an ANN can be retrained easily to deal with environmental condition. In addition, in order to handle unstable environment, an ANN can be designed to adjust is synaptic weights in real time.

4) **Evidential response**: In pattern classification, an ANN not only provides information of selected pattern but also the confidence in that made decision. Therefore, researchers latter on will be able to eliminate ambiguous information in order to improve the performance of the network classification.

5) **Contextual information**: The structure and activation state of a neural network can represent information. A single neuron is affected potentially by the global activity of all surrounding neurons in the network.

6) **Fault tolerance**: This property enables a system to continue operating properly in the cases of the failure of some components in it. Therefore, in principle, a neural network tends to exhibits a graceful degradation in performance than a catastrophic failure.

7) **Very large scale integrated (VLSI) implementability**. The parallel structure of a neural network enables its ability to compute certain tasks. In addition, this feature makes a neural network suitable for implementation in VLSI technology.

8) **Uniformity of analysis and design**: This feature exposes itself in many ways. A neuron, in one form or another, performs a common ingredient to all neural networks. Its commonality enables sharing ability and learning algorithms in many applications of neural networks. In addition, a seamless integration of modules can be built.

9) **Neurobiological analog**: The design of neural network is inspired by biological neural network. Engineers and researchers focus on neurobiology for new ideas for solving complicated problems.

3.4.3) Support Vector Machine (SVM)

a) Introduction

The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 [18]. SVM is a supervisor leaning method that maps inputs to their corresponding outputs. It can be used for classification (categorize the inputs) or a regression (estimate the desired output) [19]. For classification, the inputs are transformed to high dimensional feature spaces in which they become more separable in comparison to the original input spaces. Firstly, a hyperplane is constructed to optimally separate the training data into two classes. Then, two parallel hyperplanes (or support vectors) are constructed on each side of the hyperplane. By maximizing the distance between the two parallel hyperplanes, the data is classified into two classes. SVMs have competitive performance in many applications, such as medical diagnosis, bioinformatics, face recognition, image processing and text mining, which has used SVMs as one of the most popular, state-of-the-art tools for discovery and mining data.

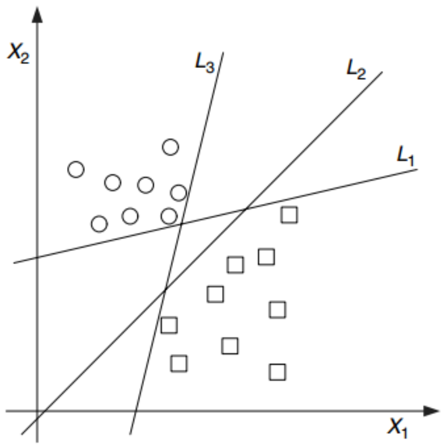


Figure 16: A training data set can be separated by many hyperplanes

b) Explanation of SVM

Let consider a linear classifier for a binary classification problem with labels y and features x. In particular, each data point x will has its corresponding class label y, which is either -1 or 1. The separating hyperplane has the form as following

$$w \bullet x - b = 0$$

where w is the normal vectors of the hyperplane, b is an offset of the hyperplane.

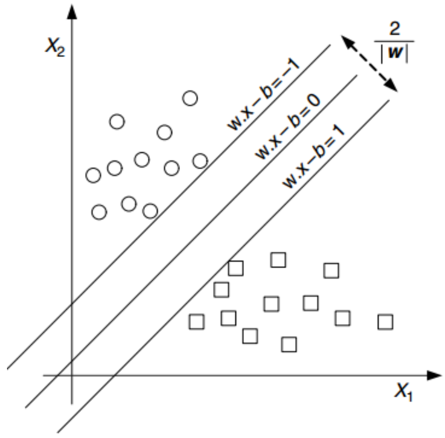


Figure 17: Schematic representation of the principle of SVM

We are focus on the maximum margin of the support vectors (the parallel hyperplans) to the optimal hyperplane. These parallel hyperplane can be described by equation

$$w \bullet x - b = 1 \text{ (for the first class)}$$

$$w \bullet x - b = -1 \text{ (for the second class)}$$

If the training data are linearly separable, we should find two hyperplanes that satisfy the condition there are no point between them and then try to maximize their distance. It means we need to ensure that all data points either

$$(w \bullet x_i - b) \geq 1 \text{ or}$$

$$(w \bullet x_i - b) \leq 1$$

this requirement can be rewritten as

$$y_i(w \bullet x_i - b) \geq 1 \text{ for } 1 \leq i \leq n$$

c) Primal form

The optimization problem is to minimize ||**w**||, however, it is difficult to solve because it depends on ||**w**||, which involves a square root. Fortunately it is possible to alter the equation by substituting ‖**w**‖ with $\frac{1}{2}||\mathbf{w}||$ without changing the solution. The factor ½ is used only for mathematical convenience. In particular, the problem is minimizing $\frac{1}{2}||\mathbf{w}||$ then substituting it into $y_i(w\bullet x_i - b) \geq 1$ to achieving solution.

d) Dual form

Writing the optimization problem in its dual form means that classification is only a function of the support vectors. The parameter w can be written as linear combination of the training data:

$$w = \sum_{j=1}^{N} a_j y_j x_j$$

Substituting w in $f(x) = w\bullet x_i - b$ and simplifying gives the final formula

$$f(x) = \sum_{i}^{N} a_i y_i (x_i{}^T x) + b$$

Comparing the value of f(x) to 1 and -1, we are able to determine whether data point is belonging to the first class (f(x) ≥ 1) or the second class (f(x) ≤ 1).

In addition, we can show the dual form of the SVM as the following optimization problem:

$$f(\alpha) = max \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i{}^T x_j$$

and subject to

$$a_i \geq 0 \; and \; \sum_{i=1}^{N} a_i y_i = 0$$

The final calculation is the optimization solution for SVM method in dual form. It provides support vectors for classification problem. These support vectors are implemented on the smartphone later on.

In the experiment, there were many SVM classifiers because SVM is the binary classification method. The Table 1 shows the information of SVM method in offline classification, while the Table 2 represents the description in online classification.

Table 1: The SVM in offline classification

	Downstairs	**Driving**	**Sitting**	**On-table**	**Upstairs**
Training set	15/120[(1)]	30/120	30/120	30/120	15/120
Testing set	10/80	20/80	20/80	20/80	10/80
Support Vectors	30	33	31	26	19

[(1)] The number of class 1 over the total number of the set.

Table 2: The SVM in online classification

	Driving	**Sitting**	**Stairs climbing**	**Walking**
Training set	4500/18000 [(1)]	4500/18000	4500/18000	4500/18000
Testing set	500/2000	500/2000	500/2000	500/2000
Support Vectors	88	87	101	108

[(1)] The number of class 1 over the total number of the set.

3.5. Metabolic equivalent of task

Metabolic equivalent of task (MET) is a common physiological measurement of the energy consumed during physical activities. It is defined as the ratio of metabolic rate at work the metabolic rate at rest [20]. One MET is equal to 3.5 ml of O_2 per kg of body weight per minute, or 1 kcal per kg of body weight per hour.

Physical activities are classified into three categories: light intensity, moderate intensity and vigorous intensity activities. Particularly, light intensity activities, which has METs smaller than 3, are sleeping, watching television, writing, desk working, and waking slow Moderate intensity activities includes bicycling, walking fast, dancing, gardening, etc. consume 3-6 METs. Activities that require more than 6 METs are vigorous intensity activities which consist of running, fast dancing, aerobics, fast swimming, etc.

In this thesis, four recognized activities are driving, sitting, stairs climbing and walking. The corresponding metabolic equivalents of those activities are 3.5, 1.3, 8 and 2.5 METs. The energy expenditure is calculated as following equation:

$$\text{Energy expenditure (kcal)} = 1.05 \times \text{METs} \times \text{duration (hour)} \times \text{weight (kg)}$$

Researchers can find more activities and their relative METs values on *Compendium of Physical Activities website* [21]. This site is designed to provide the Compendium of Physical Activities and additional resources with evidence to support the information.

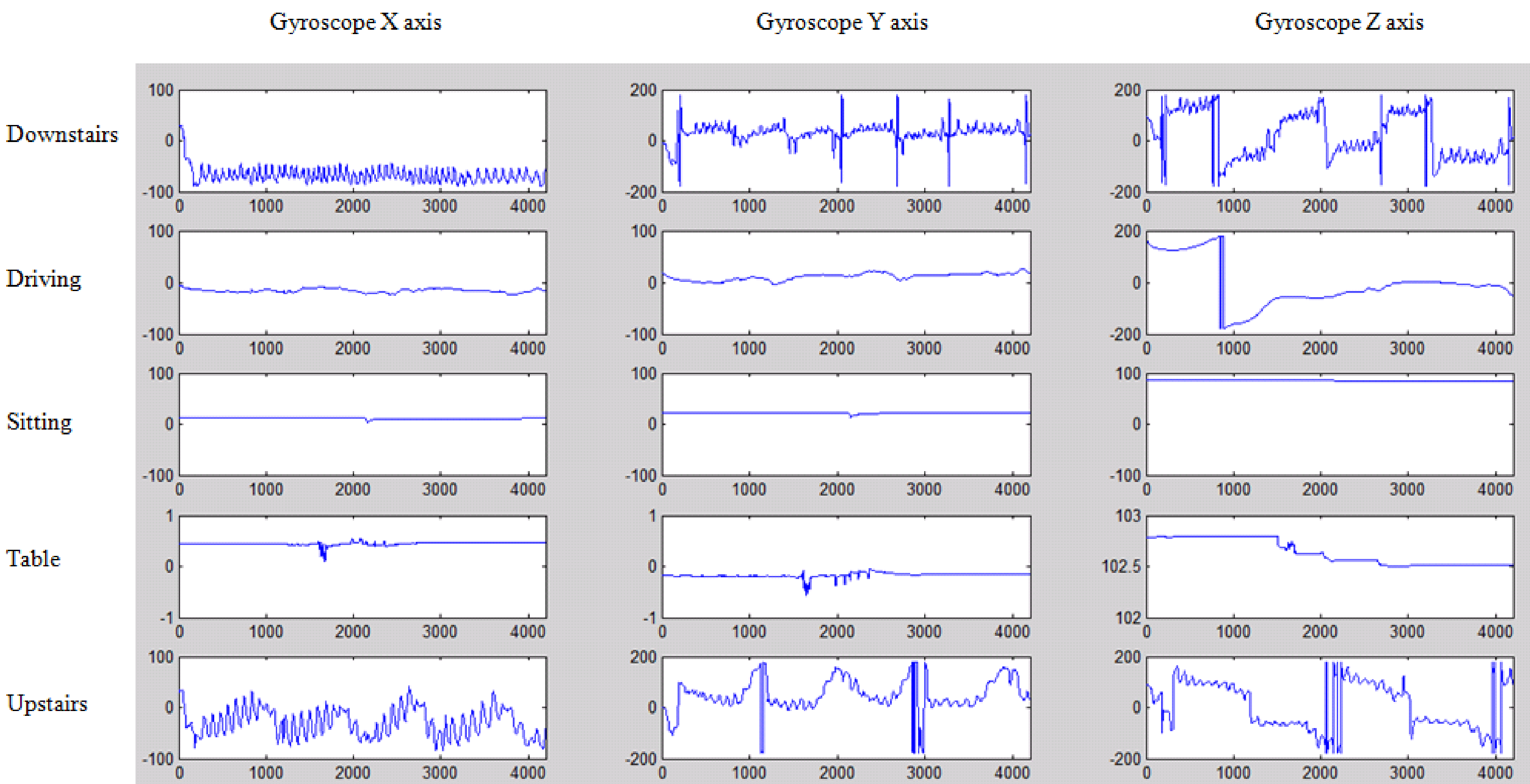


Figure 18: Example of data taken from gyroscope sensor

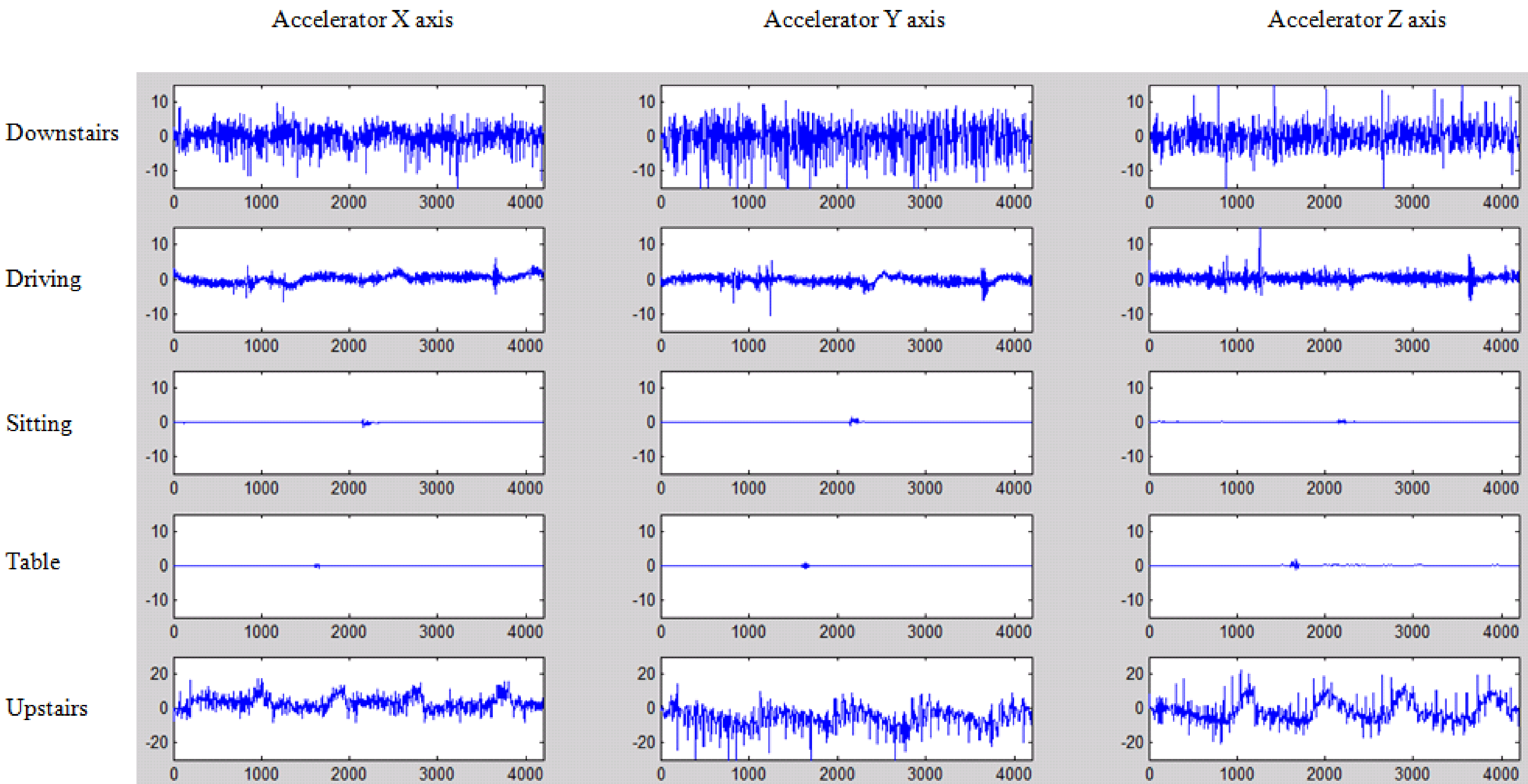


Figure 19: Example of data taken from accelerator sensor

CHAPTER 4
SMARTPHONE IMPLEMENTATION

4.1 Introduction

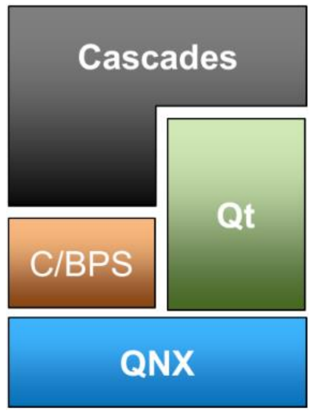


Figure 20: Blackberry 10 platform

QNX was one of the first commercially successful microkernel operating systems and was used in Blackberry 10 system. QNX system was operated in the form of many small tasks (or servers). It is used to allow developers to turn off any functionality they do not require without having to change the OS itself. Blackberry Platform Services (BPS) provides a low level software development kit for building and developing Blackberry platform applications. BPS libraries allow developer to set access to hardware and device resources. Qt is a cross-platform application framework with design Graphic User Interface (GUI), command lines, applications. In addition, it is an extension of C++ (meta object complier) for software development kit. Cascades is not only the replacement for Qt UI Creation Kit, which includes libraries to access device hardware, but also supports development tools.

4.2 QML and C++ Communication

QML is a programming language that is designed to describe the user interface (UI) of an application in both appearance and behavior. Although, it was as a simple programming language, it provides developers a powerful tool to create UIs. They can use QML to describe not only the structure but also behavior of UI components.

In QML and C++ communication, it is possible to communicate from QML to C++ and vice versa. In particular, QML can handle C++ signal, expose C++ QObject and call C++ functions. On the other hand, programmer can read and write to QML properties and call functions on QML objects by using C++.

4.2.1) Exposing C++ values into QML.

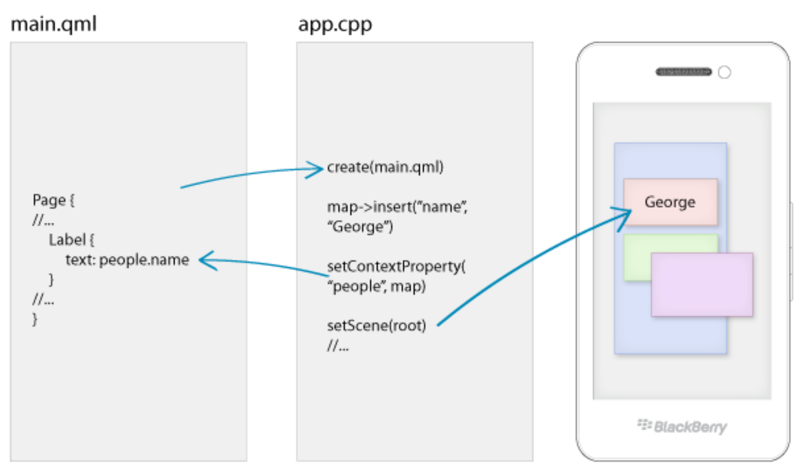


Figure 21: Exposing C++ values into QML

It is impossible to send data from C++ into QML in an application. A convenient way to do that is using the `QDeclarativePropertyMap` class. This technique allows programmers to define key-value pairs in C++ that can bind to QML properties.For example, they can use `QDeclarativePropertyMap` to pass a name and phone number to QML. The following code shows the illustration.

```
QmlDocument *qml =
QmlDocument::create("asset:///main.qml").parent(this);
AbstractPane *root = qml->createRootObject<AbstractPane>();
QDeclarativePropertyMap* propertyMap = new QDeclarativePropertyMap;
propertyMap->insert("name", QVariant(QString("Wes Barichak")));
propertyMap->insert("phone", QVariant(QString("519-555-0199")));
qml->setContextProperty("propertyMap", propertyMap);
Application::instance()->setScene(root);
```

After coding in C++ file, in QML document, programmers can access the values by using the property name and the key that is defined in C++.

```
import bb.cascades 1.3
Page {
    Container {
        Label {
            text: "Username: " + propertyMap.name
        }
        Label {
            text: "Phone number: " + propertyMap.number
        }
    } }
```

4.2.2) Exposing C++ objects into QML

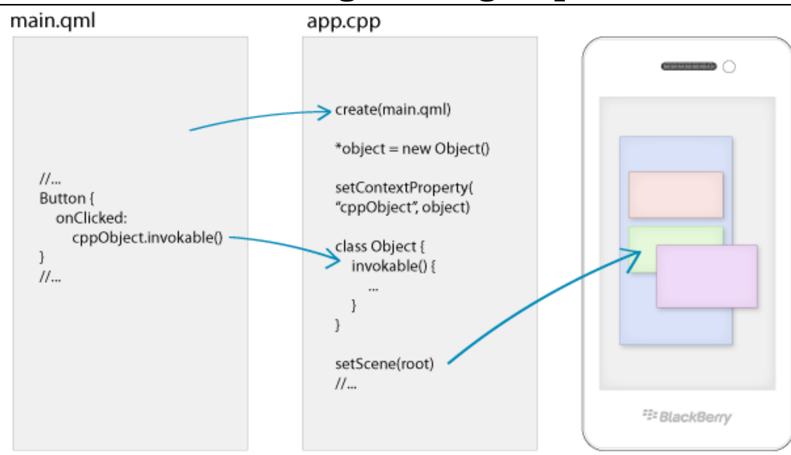


Figure 22: Exposing C++ objects into QML

It's possible to create a C++ object and expose it to QML. Before we can pass a C++ object to QML, there are some important Qt macros that we need to declare in the header file for the C++ class.

- Q_OBJECT is a required macro for classes that use the signals and slots mechanism.
- Q_PROPERTY exposes a class property to QML. It also defines the read and write values from and to the property, as well as the signal that is emitted when the property is changed.
- Q_INVOKABLE exposes member functions so that they can be invoked from QML.

QML can interact with QObjects which is declared in C++, programmers can call any Q_INVOKABLE C++ object functions. Moreover, QML can read and write

Qt properties on that QObject. The following code is an example of exposing QObject to QML document.

```
#ifndef MYCPPOBJECT_H_
#define MYCPPOBJECT_H_
#include <QObject>
#include <QMetaType>
class MyCppObject : public QObject
{
    Q_OBJECT
    Q_PROPERTY(int value READ value WRITE setValue NOTIFY
valueChanged)
public:
    MyCppObject(QObject *parent = 0);
    virtual ~MyCppObject();

    Q_INVOKABLE void reset();
    int value();
    void setValue(int i);
signals:
    void valueChanged(int);
private:
    int m_iValue;
};
#endif
```

In order to expose `MyCppClass` to QML, we create an instance of the class and use setContextProperty() to expose it. The parent is set to the `ApplicationUI` instead of using the “`this`” keyword so that when the app is destroyed, this object is as well.

```
MyCppObject *cppObject = new MyCppObject(this);
qml->setContextProperty("myCppObject", cppObject);
```

In the example below, we can access the `Q_PROPERTY` from QML by using the property name that is defined in the header file

```
Label {
    text: "Value of MyCppObject: " + myCppObject.value
}
```

4.*2.3) Using C++ classes in QML*

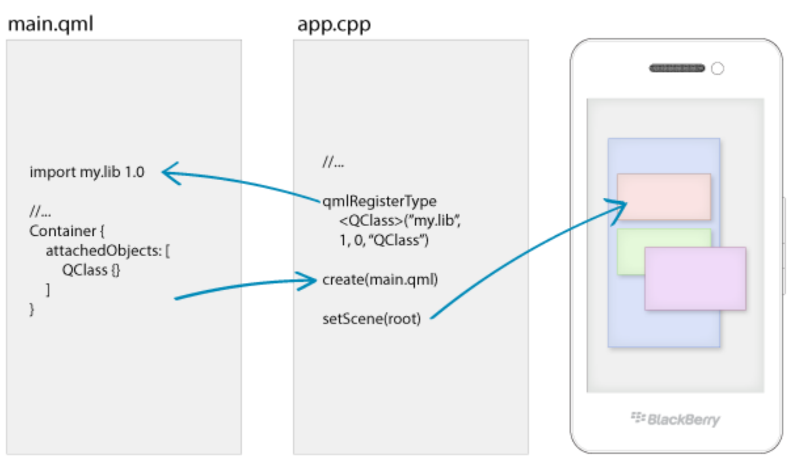


Figure 23: Using C++ classes in QML

In some cases, although we can pass C++ objects or values to QML, we may use the C++ class in QML directly. In the Cascades framework, there are a few ways to do that, whether it's a class from the core Qt library, or a class that we define yourself:

- Attach a C++ class to an existing QML component using UIObject::attachedObjects (the class must inherit QObject).

- Extend a C++ class using CustomControl and use it in QML directly.

The following example shows the way to register the QTimer class for use in QML.

```
qmlRegisterType<QTimer>("my.library", 1, 0, "QTimer");
 QmlDocument *qml =
QmlDocument::create("asset:///main.qml").parent(this);
AbstractPane *root = qml->createRootObject<AbstractPane>();
Application::instance()->setScene(root);
```

After registering the `QTimer` class, we can create and manipulate `QTimer` objects within QML document as described in the following example. At the same time the page is created, the timer is started. When the timer is complete, the text for the label is updated with new text.

```
import bb.cascades 1.0
import my.library 1.0
Page {
    Label {
        id: timerTestLabel
        text: "Hello world"
        attachedObjects: [
            QTimer {
                id: timer
                interval: 1000
                onTimeout :{
                    timerTestLabel.text = "Timer triggered"
                }
            }
        ]
    }
    onCreationCompleted: {
        timer.start();
    }
}
```

4.3 Accessing sensor data

4.3.1) Introduction

Sensors enable BlackBerry 10 device to collect information of the surrounding environment and the physical events. We use the sensors to build an application that responds to the device's position such as accelerations and rotations

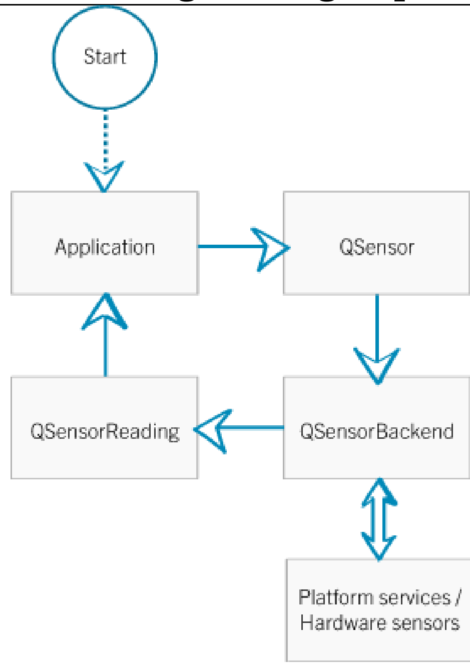


Figure 248: Sensor architecture

The sensors architecture includes QSensorReading and QSensorBackend. The QSensorReading is what we call to access data provided by the QSensorBackend. The advantage of splitting sensors into two parts is that we can use a common abstraction to access data, regardless of the sensor type. Finally, Blackberry 10 allows programmers to access sensors from QML directly, which is important feature for designing sensor-aware applications.

There are many sensors types supported by the BlackBerry 10 platform, however, in this project, we use only two sensor: accelerometer and gyroscope sensors. Accelerometer returns the device acceleration in three dimensions. Programmers can also specify which acceleration component should be reported by the sensor. For example, only the gravity component is relevant if they want to detect fallings. Gyroscope: Returns the device's angular velocity in three dimensions measured in degrees per second.

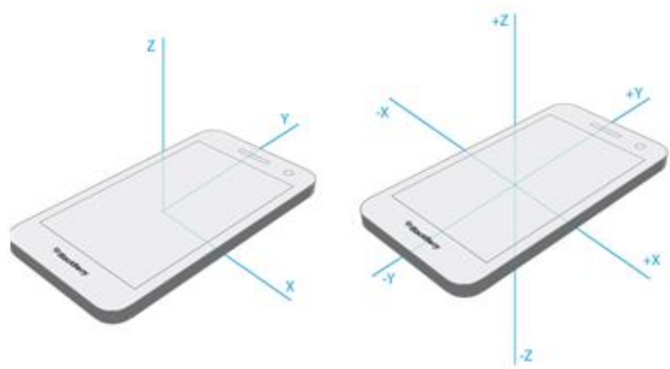


Figure 25: Accelerometer coordinate system (image source: BlackBerry website)

Unless otherwise specified, sensors use the right-hand Cartesian coordinate system. To allow for measurements in all six directions, negative values are used.

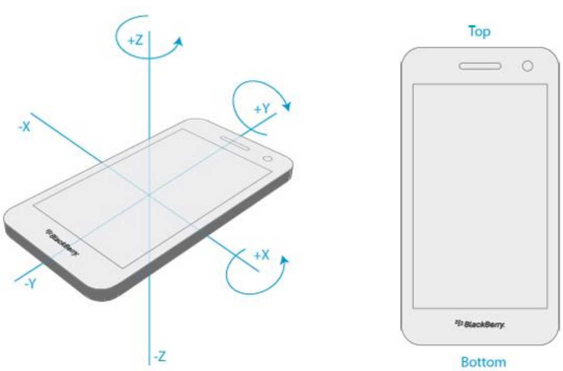


Figure 26: Gyroscope coordinate system (image source: Blackberry website)

Where rotation around an axis is used, the rotation shall be expressed as a right-hand rotation. In general, sensor data is oriented to the top of the device. If values are to be displayed on the screen, the values may need to be transformed so that they match the user interface orientation. We can show the value of sensors on the screen as shown in the figure below.

Figure 279: User interface of the application

4.3.2) Sensors in QML

Using sensors in QML is quite simple. All we need to do is declare the sensor as an attachedObject property of a control in the scene graph. The following is the code to read accelerometer values and send them in to C++.

- Accelerometer sensor

```
attachedObjects: [
    Accelerometer {
        id: alarm
        property double x: 0
        property double y: 0
        property double z: 0

        property bool movement: false
        active: true

        axesOrientationMode: Accelerometer.FixedOrientation
        accelerationMode: Accelerometer.User

        alwaysOn: true
        skipDuplicates: true

        onReadingChanged: {
            x = reading.x;
            y = reading.y;
            z = reading.z;
            _app.update_acc(x, y, z);
        }
    }
]
```

- Gyroscope sensor

```
attachedObjects: [
    RotationSensor {
        id: rotation
        property real a: 0
        property real b: 0
        property real c: 0

        active: true
        alwaysOn: true

        skipDuplicates: true
        onReadingChanged: {
            a = reading.x
            b = reading.y
            c = reading.z
            _app.update_rot(a, b, c);
        }
    }
  ]
```

CHAPTER 5
RESULTS

5.1. Offline results

5.1.1) Downstairs and upstairs walking – Quantitative evaluation

The raw signals of the accelerometer and gyroscope sensors were shown in the Fig. 4. For each axis, although they were different in gyroscope sensor (G_x, G_y, G_z), the accelerometer data (A_x, A_y, A_z) of going downstairs and upstairs were similar. Quantitative calculations were based on AR coefficient, fractal dimension, mean, and standard deviation. The result showed confusions between those activities (Table 1 and Table 2).

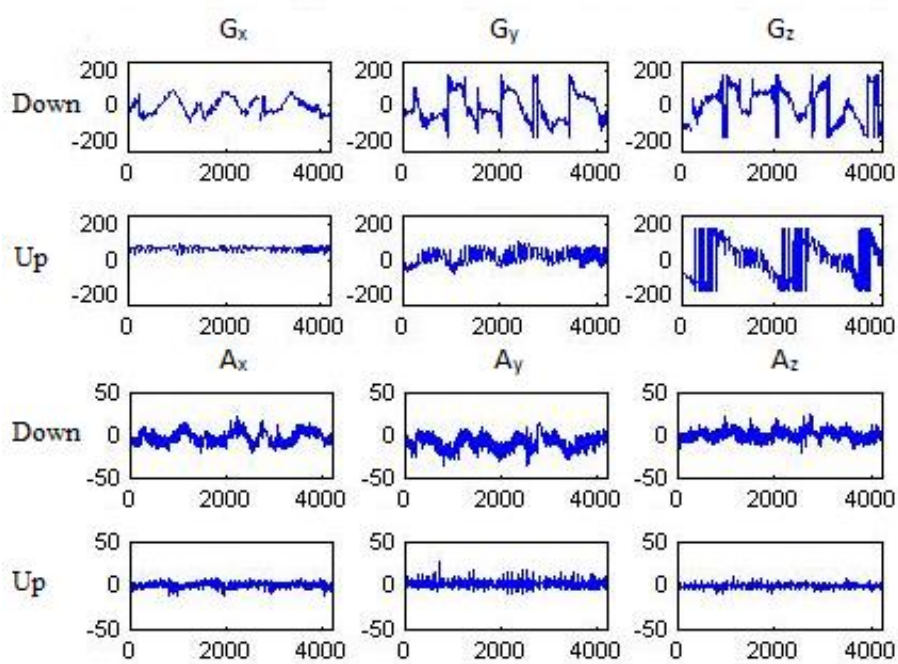


Figure 28: The raw signal of going downstairs and upstairs

5.1.2. Driving and sitting – Qualitative evaluation

Although the raw data of going downstairs and upstairs were confused, driving and sitting action were clearly differentiated by observation. The further calculation confirmed that it was possible to distingue those activities. The result of feature extraction and classification showed that the accuracy of both activities was high. For example, the accuracy of driving and sitting were 80% and 100% when the kNN method was applied. In addition, with the ANN method, the driving and sitting activities were 97% and 76% respectively.

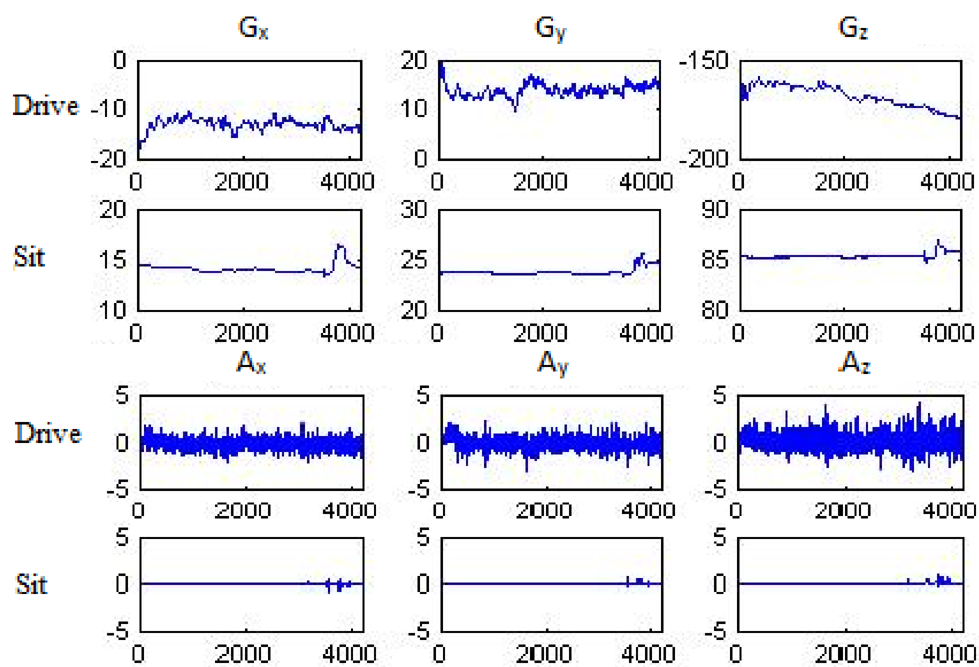


Figure 2910: The raw signal of driving and sitting

5.1.3) Classification results

❖ **kNN result**

As can be seen in Table 1, sitting with the phone in a pocket and putting it on a table had the highest degree of accuracy (100%), while the downstairs and upstairs walking were less accurate and easily confused with each other. The action of driving a motorbike had 80% accuracy and 20% false detection due to the confusion of driving with the resting and sitting action.

Table 3: The accuracy of kNN method (%)

	Downstairs	**Driving**	**Sitting**	**On-table**	**Upstairs**
Down-stairs	30	30	0	0	40
Driving	0	80	20	0	0
Sitting	0	0	100	0	0
On-table	0	0	0	100	0
Upstairs	40	0	0	0	60

❖ **ANN result**

The result shown in Table represented the level of accuracy of classification using ANN method. The accuracy of downstairs, driving, sitting, on-table and upstairs actions were 64%, 97%, 76%, 90.5%, 49% respectively. The overall accuracy was 75.3%.

Table 4: The accuracy of ANN method (%)

	Downstairs	Driving	Sitting	On-table	Upstairs
Down-stairs	64	15	4	0	17
Driving	1.5	97	1.5	0	0
Sitting	2	15	76	7	0
On-table	5	3.5	1	90.5	0
Upstairs	49	2	0	0	49

- **SVM result**

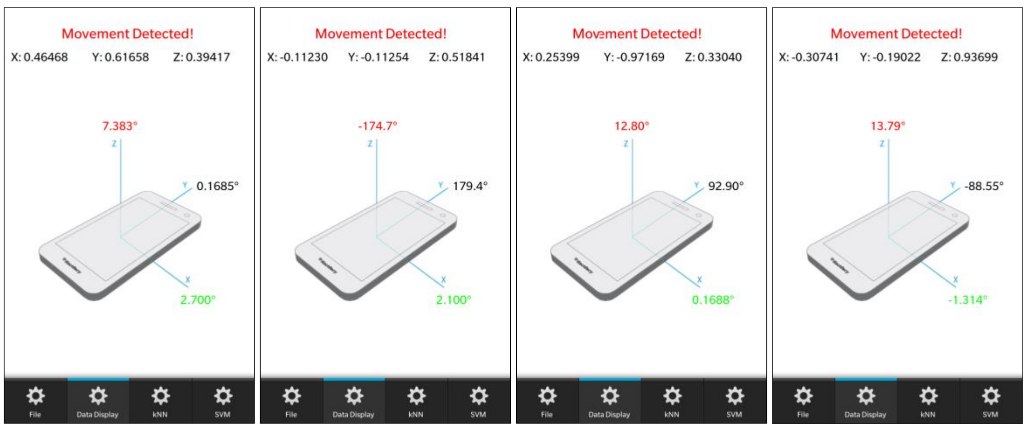
The table shows the evaluation of SVM method in classification five daily activities. It can be clearly seen that the accuracy of all activities is high. The average accuracy of SVM method is 94.5%.

Table 5: The accuracy of SVM method (%)

	PPV	Sensitivity	Specificity	F1 score	Accuracy
Down	46.2	60	90	52.2	86.3
Drive	82.6	95	93.3	88.4	93.8
Sit	100	85	100	91.9	96.3
Table	100	100	100	100	100
Up	88.9	80	98.6	84.2	96.3

5.2) Online results

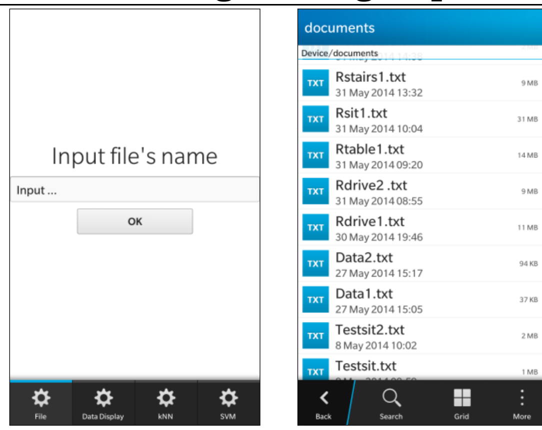
In practice, the accelerometer and gyroscope readings were combined to measure the device's displacement using six degrees of freedom. These degrees of freedom included three accelerations measured by the accelerometer and three rotations measured by the gyroscope. Different positions of the smartphone provide different values of sensors.



(A) (B) (C) (D)

Figure 30: Different positions of the smartphone: upward (A), downward (B), rightward (C), leftward (D)

When the sensors are activated, it continuously represents sensor values until the application is turned off. In order to record the coordinates of the sensors, users can save them to a text file. They can set the name of the text file in the "File" tab as can be seen in Figure 32. The Figure show the location of text files, where the history of measurement is recorded.



(A) (B)

Figure 31: The application allows users to name the text files (A) and then save them in one folder (B)

Visualizing data is a way to evaluate whether it is classifiable or not. The Figure 33 represented four activities included driving, siting, stairs climbing and walking in 3D space. Each activity is represented by 6000 data points which are sets of 6 means, 6 standard deviations and 6 fractal dimensions.

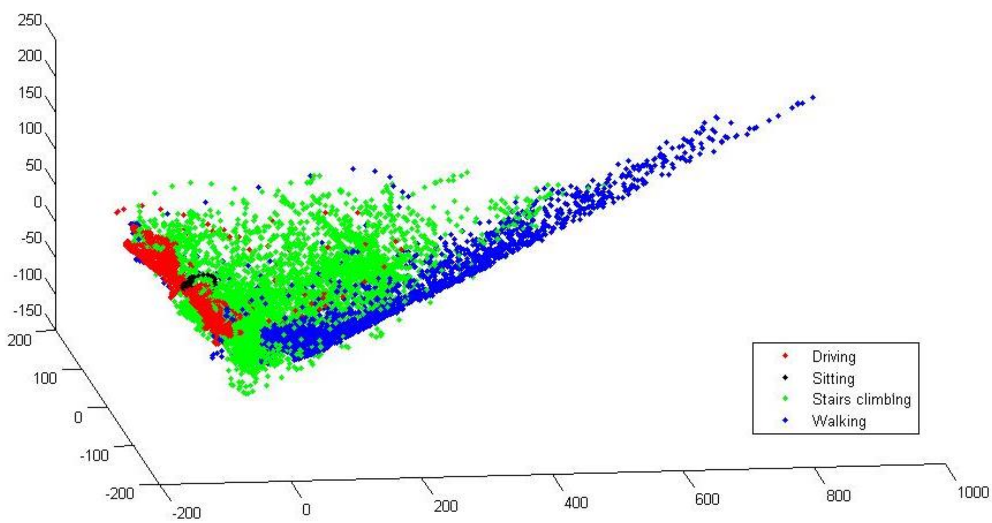


Figure 3211: The display of 4 activities in 3-dimensional space.

It is complicated and time consuming for training total 24000 data point in smartphone implementation. Therefore, it is necessary to minimize the training data before it is implemented on smartphone. Figure 34 show the general accuracy for different training sets in kNN and SVM methods.

Table 6: The accuracy of kNN and SVM for the same training sets

Training points	3	15	30	60	90	120	180	240	300	360	390
kNN	70.5	85.37	87.4	90.24	91.86	92.92	94.27	95.07	95.79	96.22	96.47
SVM	57.57	68.52	82.56	88.62	90.87	91.46	93.26	93.86	94.68	95	95.37

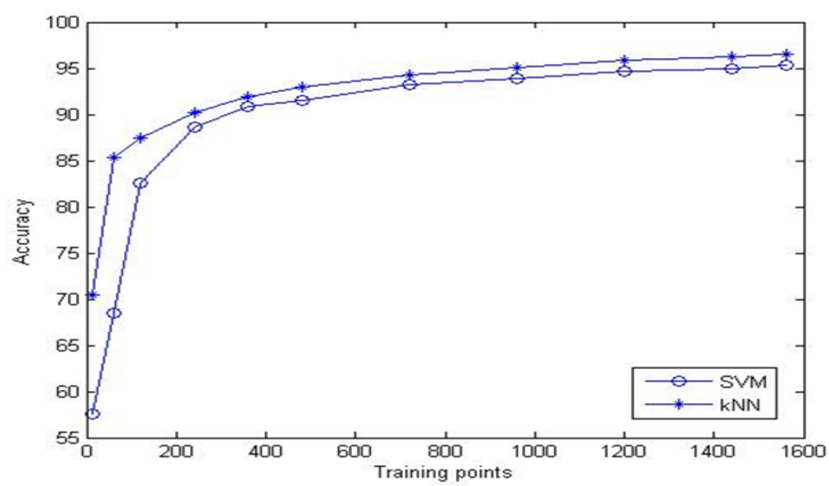


Figure 3312: The accuracy of different training sets in kNN and SVM methods

In smartphone implementation, I decide to apply a training set of 240 points for both kNN and SVM classification. The corresponding accuracy is 90.24% for kNN classifier and 88.62 % for SVM classifier. The subject can hold the smartphone in his hand or put it in his pocket when he doing activities. The data is classified automatically and display results on the smartphone interface as Figure 35.

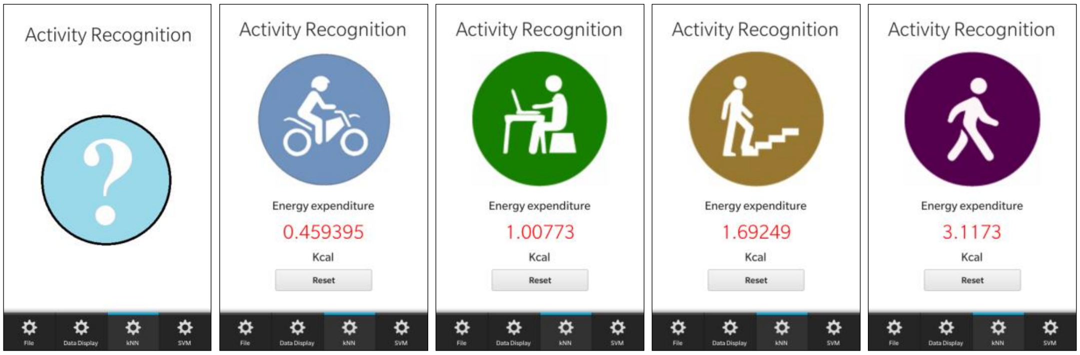


Figure 34: The smartphone interfaces of unclassified activity (A), driving (B), sitting (C), stairs walking (D) and walking (E)

The red numbers emphasize the energy expenditure in kilocalories of users during activities. Each activity will have different metabolic equivalents. In details, the METs of driving, sitting, stairs climbing and walking in general are 2.5, 1.3, 8.0, 2.5 respectively. There is a “Reset” button to allow users reset the energy expenditure value to 0.

CHAPTER 6
CONCLUSION AND DISCUSSION

This paper has focused on the daily activity recognition by analyzing data of accelerometer and gyroscope sensors of a smartphone. The results showed that by using just a smartphone, we could recognize various basic activities such as driving motorbike, going stairs, sitting with a smartphone in pocket and putting the smartphone on a table.

In both methods of classification including kNN and ANN, the results of recognizing driving, sitting and putting the phone on a table were considerably accurate. It ranged from 76 to 100 percentages. On the other hand, poor recognition results were presented for going downstairs and upstairs activities due to the ambiguity of accelerometer and gyroscope data. Therefore, those two activities should be concerned as a single activity for better recognition results.

In smartphone implementation, among three classifiers, kNN and SVM are methods that are applicable on smartphone because of several reasons. The first reason is they are not only simple but also effective classification methods. In addition, they can be implemented and updated very easily in order to adapt with the improvement of training set. Although ANN is also an effective classification method, it is complicated to be implemented on the smartphone. Therefore, kNN and SVM have used as useful classifiers.

This application is design to improve the lifestyle of users. This can measure durations of activities in their daily lives such as driving, sitting, stairs climbing and walking. The result of classification and the duration of activity are used to calculate the energy expenditure. This energy is good evidence as well as a persuasive message to the user about their lifestyle. Therefore, the users can improve their lifestyle in order to have a better health.

Although the smartphone can recognize activities with high accuracy, my thesis still have limitations. Firstly, there is a little different in the way people hold the smartphone in their hand or keep it in their pocket. Therefore, the accuracy of activity recognition may vary in comparison with the accuracy in this thesis. Secondly, the activities were recorded separately and independently. As the result of this, the subject needs to perform activities for a while in order to get right recognitions. Additionally, when the subject changes from one activity to another, the recognition will be confused for a short time. This limitation will be eliminated in the future work.

In the future work, I intend to focus on the logfile of the experiences. Logfile record activities in order to provide information that can be used to understand the principle of complex activities, particularly in the activity history management and extension of activities. The next goal I have planned to achieve is working with sequence of activities. In details, I will pay attention to the changes from one activity to others logically. Therefore, the activities recognition will logically and persuasive.

REFERENCES

[1] Department of Health Statistics and Informatics in the Information "Global health risks: mortality and burden of disease attributable to selected maor risks", Geneva, World Health Organization, 2009, pp 11.

[2] US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Chronic Disease Prevention and Health Promotion, "Physical activity and health: a report of the Surgeon General.Atlanta", 1996. pp.9-14.

[3] Qi A., Ishikawa Y., Nakagawa, Kuroda A., Oka H., and Yamakawa H., et al., "Evaluation of wearable gyroscope and accelerometer sensor (PocketIMU2) during walking and sit-to-stand motions," in Roman, 2012, pp. 731-736.

[4] J.-S. Wang, F.-C. Chuang, and Y.-T. Yang, "A Wearable Physical Activity Sensor System: Its Classification Algorithm and Performance Comparison of Different Sensor Placements," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. vol. 6839, D.-S. Huang, Y. Gan, P. Gupta, and M. M. Gromiha, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 447-454.

[5] K. Altun and B. Barshan, "Human Activity Recognition Using Inertial/Magnetic Sensor Units," in *Human Behavior Understanding*. vol. 6219, A. Salah, T. Gevers, N. Sebe, and A. Vinciarelli, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 38-51.

[6] T. Grosse-Puppendahl, E. Berlin, and M. Borazio, "Enhancing Accelerometer-Based Activity Recognition with Capacitive Proximity Sensing," in *Ambient Intelligence*. vol. 7683, F. Paternò, B. Ruyter, P. Markopoulos, C. Santoro, E. Loenen, and K. Luyten, Eds., ed: Springer Berlin Heidelberg, 2012, pp. 17-32.

[7] J. Pansiot, D. Stoyanov, D. McIlwraith, B. L. Lo, and G. Z. Yang, "Ambient and Wearable Sensor Fusion for Activity Recognition in Healthcare Monitoring Systems," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*. vol. 13, S. Leonhardt, T. Falck, and P. Mähönen, Eds., ed: Springer Berlin Heidelberg, 2007, pp. 208-212.

[8] Brezmes T., Gorricho , and J. Cotrina, "Activity Recognition from Accelerometer Data on a Mobile Phone," in Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. vol. 5518, 2009, pp. 796-799.

[9] Yabo C., Yujiu Y., and Wenhuang L., "E-FallD: A fall detection system using android-based smartphone," in Fuzzy Systems and Knowledge Discovery (FSKD), 9th International Conference on, 2012, pp. 1509-1513.

[10] Dantzig S., Geleijnse G., and Halteren A., "Toward a persuasive mobile application to reduce sedentary behavior," Personal and Ubiquitous Computing, vol. 17,2013, pp. 1237-1246.

[11] Open source framework Native SDK for Blackbery 10. [Online]. Available: http://developer.blackberry.com/native/

[12] J. L.Semmlo, Biosignal and Biomedical Image Processing MATLA B-Based Applications, Marcel Dekker, 2004.

[13] Higuchi, T., Relationship between the fractal dimension and the power law index for a time series: a numerical investigation,Physica D,46, 1990,pp.254--264.

[14] Higuchi T., "Approach to an irregular time series on the basis of the fractal theory," Physica D: Nonlinear Phenomena, vol. 31, 1988, pp. 277-283

[15] P. W. Jones, A. Osipov, and V. Rokhlin, "A randomized approximate nearest neighbors algorithm," A*pplied and Computational Harmonic Analysis*, *v*ol. 34, pp. 415-444, 5// 2013.

[16] "Artificial Neural Networks," in E*ncyclopedia of Machine Learning*, C. Sammut and G. Webb, Eds., ed: Springer US, 2010, pp. 44-44.

[17] S. Haykin, N*eural Network A Comprehensive Foundation*, 2 ed.: Pearson Education, 1999.

[18] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," presented at the Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, USA, 1992.

[19] D. Olson and D. Delen, "Support Vector Machines," in A*dvanced Data Mining Techniques*, ed: Springer Berlin Heidelberg, 2008, pp. 111-123.

[20] A*merican College of Sports Medicine, ACSM'S metabolic calculations handbook*, 2006.

[21] Ainsworth BE, Haskell WL, Herrmann SD, Meckes N, Bassett Jr DR, Tudor-Locke C, Greer JL, Vezina J, Whitt-Glover MC, Leon AS. The Compendium of Physical Activities Tracking Guide. Healthy Lifestyles Research Center, College of Nursing & Health Innovation, Arizona State University. Retrieved [date] from the World Wide Web.
https://sites.google.com/site/compendiumofphysicalactivities/

APPENDIX

Human Activity Recognition and Monitoring Using Smartphones

Vu Ngoc Thanh Sang[1], Nguyen Duc Thang[1], Vo Van Toi[1], Nguyen Duc Hoang[2], Truong Quang Dang Khoa[3]

[1] Department of Biomedical Engineering, International University - VNU-HCMC, Vietnam
[2]Posts and Telecommunications Institute of Technology-HCM, Vietnam
[3]Tokyo University of Agriculture and Technology, Japan

Abstract — **The sedentary lifestyle is becoming popular especially for intellectual works. Although a physical inactivity lifestyle may cause many unexpected illnesses, it is complicated to build up a positive lifestyle due to the lack of reminder systems to manage and monitor physical activities of people. This research represents an effective way for daily activity monitoring using accelerometer and gyroscope sensors embedded in a smartphone. Signals were recorded from accelerometer and gyroscope sensors while a user was wearing the smartphone and performing different activities (going downstairs, going upstairs, sitting with the phone in a pocket, driving and putting the phone on the table). The classification algorithms with k-nearest-neighbor (kNN) and artificial neural network (ANN) were applied to recognize user's activities. The overall accuracy of recognizing five activities was 74% for kNN and 75.3% for ANN respectively. Based on the recognized activities during a day, users are able to manage their daily activities for a better life.**

Keywords — ***Smartphone, activity recognition, activity monitoring.***

I. INTRODUCTION

As reported by the World Health Organization, 3.2 million annual deaths were associated with physical inactivity lifestyle[22]. In order to have a better health, people at all ages should practice physical activities at least thirty minutes per day according to United States Department of Health and Human Services [23]. These activities may include gardening, bicycling, fast dancing, two-mile walking, stairs walking, and playing sports, etc. Such regular physical activities can decrease premature death by preventing heart diseases, reducing diabetes risks and controlling high blood pressure. However, despite the great benefits that physical activities offer, it is complicated to build up a good habit to achieve the desired healthy lifestyle. One of the major obstacles is that most people are too busy to spend at least thirty minutes per day to perform physical exercises. The best they can do is to practice any activities whenever it is possible, leading to a problem that they cannot manage duration as well as intensity of those physical activities.

Different approaches have been proposed to monitor users' activities by using accelerometer and gyroscope sensors. They can be divided into two major approaches: wearable sensor based-approaches and smartphone-based approaches. In the wearable sensor-based approaches, three gyroscope and accelerometer sensors were used to measure joint angles including hip, knee and ankle joints of monitoring subjects for walking, sit-to-stand, and stand-to-sit activities[24]. Another sensor-based approach was developed by Electronic Cervical Range of Motion Measurement System (ECROM)[25]: A three-axis accelerometer was used as a tilt sensor to measure both the frontal and extension of the cervical spine while a resonator gyroscope played a role as a rotation sensor to measure the transverse of the cervical spine. Alternatively, with the widespread of smartphones nowadays, many researchers have currently focused on existing sensors embedded in a smartphone for the task of recognizing users' activities, e.g. the implementation of real time activity recognizer[26]. The achieved accuracies ranged from 70% to 90% for different activities consisting of walking, climbing downstairs, climbing upstairs, sitting down, standing up, falling and so forth. In another aspect, a smartphone was used to detect falls in human [27]: Subjects were asked to repeat both falls as well as normal actions such as walking, sitting down and jumping to generate training dataset. Observing differences of the two activities, researchers investigated adaptive thresholds to distinguish them in a feature space. Utilizing a smart phone as a reminder to ask people to take regular breaks from prolonged sitting seems promising since it motivates a mobile-phone user to become more active. In an application[28], a smartphone generated acoustic warnings and vibration with virtual persuasive messages to remind a user to take a break.

In this paper, we have proposed a new method using a smartphone to recognize daily activities of a user. The recognized activities include going downstairs, going upstairs, sitting, driving and putting the phone on the table. For going downstairs, going upstairs, sitting and driving activities, the smartphone was simply considered to be in user's pocket. Obviously such activities are mainly found by an individual during a working day, thus our approach plays an important role to evaluate how much energy a user has spent during a day from which he or she can adjust their lifestyle accordingly.

II. METHODS

The overall system was described in Fig. 1 where the signals from a accelerometer and a gyroscope of a mobile phone were preprocessed and inputted into Feature extraction module to extract main features. Certain characteristics of signals such as auto regressive (AR) coefficients, fractal dimension, mean and standard deviation were calculated. The features were later classified with k-nearest neighbor (k-NN) and artificial intelligence network (ANN) to recognize different activities.

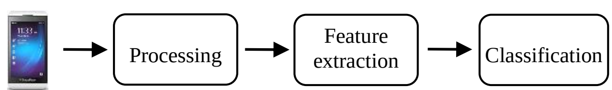


Fig. 1 The process of classifying signals.

A. Device

a) Hardware

The system was implemented on the infrastructure of Blackberry Z10, STL100-2 with a dual-core central processing unit (CPU) running at 1.5 GHz, 2 GB RAM and 16GB storage. The phone is 130 x 65.6 x 9 mm in size and 137.5 grams in weight. The touch screen has a 768 x 1280 resolution. Most importantly, the phone supports a accelerometer, gyroscope, proximity and compass sensors. However, this paper used only accelerometer and gyroscope sensors to record motions from users. Electronic accelerometer and gyroscope sensors have influenced effectively physical activity detection in living subjects since the accelerometer respond to changing velocities of subject's movements and the gyroscope sensor responded to changing orientation of a subject's movements.

b) Software

Our application was developed on the Blackberry 10 Native Platform. Blackberry 10 Native is an open source framework designed for mobile devices with a powerful Software Development Kit (SDK) based on the combination of C++ and Qt/QML framework[29]. Blackberry 10 Native SDK provides necessary drivers needed to interact with the hardware and libraries needed to deploy a smartphone application. Furthermore, Blackberry 10 Native allows programmers to access accelerometer and gyroscope sensors and to save their values into a log for further analyses.

B. Processing

Our approach required two recorded datasets for training and testing the classifier. The activity labels of both training and testing were stored in a database, hence the accuracy of our approach was evaluated. We recorded accelerometer and gyroscope sensors' values and the corresponding time index. The accelerometer was designed to detect changes due to movements and vibrations of the phone. The gyroscope sensor was used to measure orientation and rotation of the phone. In the processing stage, total six variables were shown on the screen and sent to a storage at the same time. The taken time for recording those variables was a period of 60 seconds. Different activities were saved into corresponding text files with their labels. There were five labeled activities including going downstairs, going upstairs, driving, sitting, and putting the phone on the table.

C. Feature extraction

a) Auto regressive coefficient (AR)

The most common model used to approach spectrum estimation of random signals is AR. In AR model, the data output is assumed to be generated by its previous values by the following input – output equation:

$$y[n] = \sum_{k=1}^{p} a_k y[n-k] + \varepsilon[n] \tag{1}$$

where $y[n]$ is the sum of the relation constant of $y[n\text{-}k]$ as the previous values, a_k is the coefficients at the lag k, $\varepsilon[n]$ is a zero mean white noise process with unknown variance and p is the order of the system.

b) Fractal dimension

Fractal dimension presents the irregular characteristic of a time series based on the power law index[30]. Although the fractal dimension can be considered similar to Euclidian dimension, there were significant differences between them. The main difference lied in the fact that fractal dimension may be a non-integer. The Euclidian dimension represents a mathematical system of a set, while the fractal dimension demonstrates how a set in micro-scale is similar to its counterpart in the macro-scale. The closest integer value of a fractal dimension shows its equivalence to the Euclidian dimension. For example, if the fractal dimension is close to zero, it describes a point (0-dimension in Euclidian). It describes lines when it approaches one (1-dimension in Euclidian) and surfaces when it approaches to two (2-dimension in Euclidian), and so on.

Higuchi's algorithm is an effective method to calculate the fractal dimension in time domain[31]. It is based on the calculation of the length of the curve in time series. The length $L_m(k)$ of the curve has k-segments is calculated by the following formula:

$$L_m(k) = \frac{1}{k}\left\{\left(\sum_{i=1}^{\left[\frac{N-m}{k}\right]} \left|X(m+ik) - X(m+(i-1)k)\right|\right)\frac{N-1}{\left|\frac{N-m}{k}\right|k}\right\} \quad (2)$$

where N is the number of data samples, m is an integer smaller than k $(m = 1,2,\ldots,k)$. The relationship of the length $L_m(k)$ and the fractal dimension D is shown as the following formula:

$$\langle L(k)\rangle \sim k^{-D} \quad (3)$$

where $\langle L(k)\rangle$ is the average value over k set of $L_m(k)$.

c)Mean

The use of mean is usually for measuring central tendency. Without weights assigned on data, mean is referred to the arithmetic mean. The arithmetic mean $\bar{x}$ is equal to the sum of all variables x_n divided by the number of variables N and calculated as follow:

$$\bar{x} = \frac{1}{N}\sum_{n=0}^{N} x_n \quad (4)$$

d) Standard deviation

The standard deviation is the measurement of variability evaluating a range of the data set and the relationship of the mean with the rest of the data. The standard deviation will be small if the data points are close to the mean. On the other hand, if many data points are far from the mean, it means that the variance of responses is wide and the standard deviation is large. If data values are equal at all points, then the standard deviation is zero. In addition, the standard deviation is used to predict the power of the signal and calculated using the following formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \quad (5)$$

where $\bar{x}$ is the mean of data and N is the number of data points.

D. Classification

a) k-Nearest Neighbor (kNN)

Firstly, the non-parametric kNN algorithm is used for classification. In this research, the neighbors were taken from five classes (going downstairs, going upstairs, sitting, driving, walking and putting the phone on the table). Recorded data with class labels was divided into two sets: training set (classified) and testing set (unclassified). For the recognition of the testing set, the Euclidian distance[15] was computed. Here, the Euclidean distance is the distance between points $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ in a feature space and calculated as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}\,. \quad (6)$$

In the next step, the distance from each sample in the testing set to all the samples in training set is calculated and the minimum value is chosen. Therefore,the classified label is based on the nearest distance of each testing point to the corresponding class in the training set.

b) Artificial Neuron Network (ANN)

ANN is defined as a computational model based on biological neural networks[16]. Fig.2 shows the model of a neuron which forms an ANN. There are three basic elements of a neural model: weight, summation and activation function. An input signal x_j connected to neuron k is multiplied with its own weight w_{kj}. The summation module adds all the input signals together and combines the result to the bias b_k. The activation potential v_k is the summation of the inputs with weights and bias. The activation function is the function that limits the amplitude of a output of a neuron. Common activation functions are threshold, piecewise and sigmoid functions.

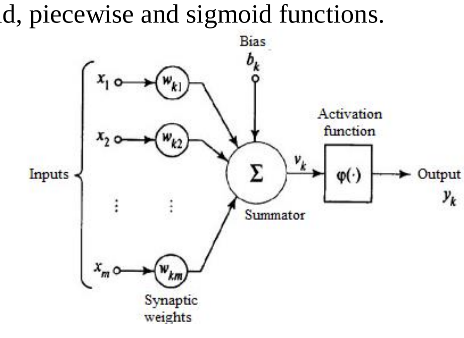


Fig. 2 The neural model.

An ANN has three layers described as input, hidden and output layers. The input layer receives signals from external environments or other networks. The hidden layer intervenes inputs and outputs in several useful manners. The more added hidden layers, the higher order statistic the network is enabled to extract. The output layer constitutes an overall response of the network to the activation pattern.

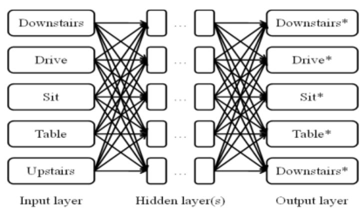


Fig. 3 The artificial neural network layer.

IV. RESULTS

A. Downstairs and upstairs walking – Quantitative evaluation

The raw signals of the accelerometer and gyroscope sensors were shown in the Fig. 4. For each axis, although they were different in gyroscope sensor (G_x, G_y, G_z), the accelerometer data (A_x, A_y, A_z) of going downstairs and upstairs were similar. Quantitative calculations were based on AR coefficient, fractal dimension, mean, and standard deviation. The result showed confusions between those activities (Table 1 and Table 2).

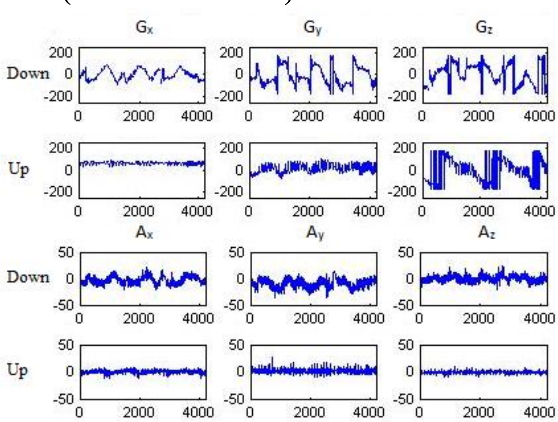


Fig. 4 The raw signal of going downstairs and upstairs.

B. Driving and sitting – Qualitative evaluation

Although the raw data of going downstairs and upstairs were confused, driving and sitting action were clearly differentiated by observation. The further calculation confirmed that it was possible to distingue those activities. The result of feature extraction and classification showed that the accuracy of both activities was high. For example, the accuracy of driving and sitting were 80% and 100% when the kNN method was applied. In addition, with the ANN method, the driving and sitting activities were 97% and 76% respectively.

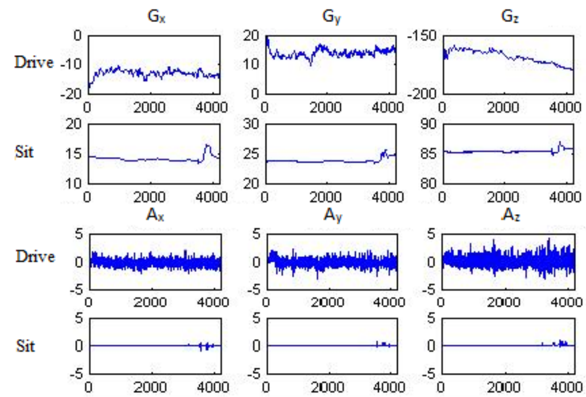


Fig. 5 The raw signal of driving and sitting.

C. General result

As can be seen in Table 1, sitting with the phone in a pocket and putting it on a table had the highest degree of accuracy (100%), while the downstairs and upstairs walking were less accurate and easily confused with each other. The action of driving a motorbike had 80% accuracy and 20% false detection due to the confusion of driving with the resting and sitting action.

Table 1: The accuracy of kNN method (%)

	Downstairs	Driving	Sitting	On-table	Upstairs
Down-stairs	30	30	0	0	40
Driving	0	80	20	0	0
Sitting	0	0	100	0	0
On-table	0	0	0	100	0
Upstairs	40	0	0	0	60

The result shown in Table 2 represented the level of accuracy of classification using ANN method. The accuracy of downstairs, driving, sitting, on-table and upstairs actions were 64%, 97%, 76%, 90.5%, 49% respectively. The overall accuracy was 75.3%.

Table 2: The accuracy of ANN method (%)

	Downstairs	Driving	Sitting	On-table	Upstairs
Down-stairs	64	15	4	0	17
Driving	1.5	97	1.5	0	0
Sitting	2	15	76	7	0
On-table	5	3.5	1	90.5	0
Upstairs	49	2	0	0	49

V. CONCLUSION AND DISCUSSION

This paper has focused on the daily activity recognition by analyzing data of accelerometer and gyroscope sensors of a smartphone. The results showed that by using just a smartphone, we could recognize various basic activities such as driving motorbike, going stairs, sitting with a smartphone in pocket and putting the smartphone on a table.

In both methods of classification including kNN and ANN, the results of recognizing driving, sitting and putting the phone on a table were considerably accurate. It ranged from 76 to 100 percentages. On the other hand, poor recognition results were presented for going downstairs and upstairs activities due to the ambiguity of accelerometer and gyroscope data. Therefore, those two activities should be concerned as a single activity for better recognition results in future.

REFERENCES

1 Department of Health Statistics and Informatics in the Information "Global health risks: mortality and burden of disease attributable to selected maor risks", Geneva, World Health Organization, 2009, pp 11.

2. US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Chronic Disease Prevention and Health Promotion, "Physical activity and health: a report of the Surgeon General.Atlanta", 1996. pp.9-14.

3. Qi A., Ishikawa Y., Nakagawa, Kuroda A., Oka H., and Yamakawa H., *et al.*, "Evaluation of wearable gyroscope and accelerometer sensor (PocketIMU2) during walking and sit-to-stand motions," in Roman, 2012, pp. 731-736.

4. Lian E., Hachadorian J., Hoan N., and Toi V., "A Novel Electronic Cervical Range of Motion Measurement System," in *The Third International Conference on the Development of Biomedical Engineering in Vietnam*. vol. 27, V. Toi and T. Khoa, Eds., ed: Springer Berlin Heidelberg, 2010, pp. 140-143.

5. Brezmes T., Gorricho , and J. Cotrina, "Activity Recognition from Accelerometer Data on a Mobile Phone," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. vol. 5518, 2009, pp. 796-799.

6. Yabo C., Yujiu Y., and Wenhuang L., "E-FallD: A fall detection system using android-based smartphone," in *Fuzzy Systems and Knowledge Discovery (FSKD), 9th International Conference on*, 2012, pp. 1509-1513.

7. Dantzig S., Geleijnse G., and Halteren A., "Toward a persuasive mobile application to reduce sedentary behavior," *Personal and Ubiquitous Computing*, vol. 17,2013, pp. 1237-1246.

8. Open source framework Native SDK for Blackbery 10. [Online]. Available: http://developer.blackberry.com/native/

9. Higuchi, T., Relationship between the fractal dimension and the power law index for a time series: a numerical investigation,Physica D, 46, 1990, pp.254 -264.

10. Higuchi T., "Approach to an irregular time series on the basis of the fractal theory," *Physica D: Nonlinear Phenomena*, vol. 31, 1988, pp. 277-283.

11. Jones P.W., Osipov A., and Rokhlin V., "A randomized approximate nearest neighbors algorithm," *Applied and Computational Harmonic Analysis*, vol. 34, 2013, pp. 415-444.

12. Sammut C. and Webb G. "Artificial Neural Networks," in *Encyclopedia of Machine Learning*, , Eds., ed: Springer US, 2010, pp. 44-44.

Author: Nguyen Duc Thang
Institute: Department of Biomedical Engineering, International University
Street: Quarter 6, Linh Trung ward, Thu Duc district
City: Ho Chi Minh
Country: Vietnam
Email: ndthang@hcmiu.edu.vn