Lift and Staircase detection using smartphone sensor

Mehedi Rahman (162 0647 042)

Ahmed Zisan (161 1243 042)

Sumaia Rahman (143 0058 042)

Md. Sadman-Al-Islam (1610503042)

Department of Electrical & Computer Engineering

Date: 23/12/2019

CSE 299 Section: 12 Group: 04

Instructor Name: Mohammad Ashrafuzzaman Khan

Abstract

Activity recognition by collecting motion data from smartphone sensors has become a prominent research area in recent years. In our project, we tried to implement a system to detect if a person is using a lift or a staircase. We used smartphone sensors to collect data from our university lift and staircase. We used machine learning tools to analyze the data and provide an optimal solution to differentiate between these two activities.

Table of Contents

List of Figures

1. Introduction

In modern smartphones nowadays, accelerometer and gyro sensors are seen quite often for making them useful for various functions like motion detection, etc. An accelerometer measures the linear acceleration of movement, while a gyro, on the other hand, measures the angular rotational velocity. [1] Both the sensors measure the rate of change, but they measure the rate of change for different things. An accelerometer will detect the directional movement of a device. But it will not be able to resolve its lateral orientation or tilt during that movement accurately. A gyro can solve this problem. We planned to use both the sensors to collect our dataset.

We have also used Support vector machine(SVM) here to classify our dataset. Support vector machine technique creates a hyperplane in boundless dimensional space, which is classification and regression. Separation is accomplished by the hyperplane that has the most significant classification for the nearest training data point of any class. This nearest data point is known as a functional margin. The generalization error of the SVM classifier depends on the size of the functional margin. The SVM training algorithm builds a model based on functional margin; the category of functional margin makes a non-probabilistic binary linear classifier. This technique is a supervised learning model used for linear and non-linear classification. The non-linear classification is performed using the kernel-based function for mapping input into high dimensional feature space. [2]

In this report, we have shown a comprehensive discussion over how to distinguish between the two states between the staircase and lift with the use of already installed sensors on our phone.

2. Background

The use of accelerometer and gyro sensor is really widespread, especially fitness devices and for device protection. Before we dive into details, let us review some key concepts of our project.

2.1. Accelerometer

Accelerometer sense the acceleration of smartphones. There are three axes that are predefined. The data collected from the accelerometer is the acceleration of each axis in the units of gravitational force. A timestamp is returned together with the three axes readings. We can choose a sampling rate so that the best sampling rate can be chosen for more accuracy. The accelerometer is profoundly used in smartphone sensors-based activity recognition.

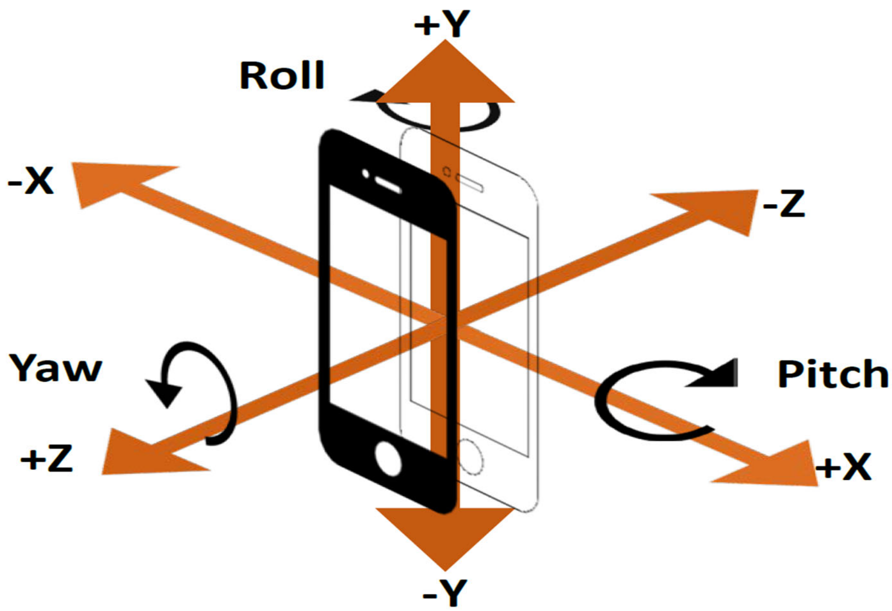

Fig. 1. Accelerometer and Gyroscope axes on Smartphones

2.2 Gyroscope

The gyroscope measures the phone's rotation rate by detecting the roll, pitch, and yaw motions of the smartphones along the x, y, and z-axis, respectively. [3]. The data collected from a gyroscope sensor is the rate of the rotation in rad/s around each axis. Gyroscope is helpful in the navigation applications as well as some smartphone games which use the rotation data. In activity recognition research, the gyroscope is used to assist the mobile orientation detection.

3. Methodology

In this section, we will discuss the method we followed to implement our project, including data collection, data preparation, graph analysis, classification, etc.

3.1 Data Collection

We collected our data through an app. We made an android app to access the accelerometer and gyroscope of any android phone and store the data in a CSV file. The three axes' data is stored in three different columns. Sensor name and timestamp are also stored with each data.

	A	B	C	D	E	F	G	H	I
1	2188298785418408; ACC; 0.112681; 4.124386; 9.864791; 0.000000; 0.000000; 0.000000								
2	2188298786269408; GYRO; -0.057805; -0.657535; 0.032847; 0.000000; 0.000000; 0.000000								
3	2188298805271533; ACC; 0.528477; 4.025036; 9.890812; 0.000000; 0.000000; 0.000000								
4	2188298806122533; GYRO; 0.001710; -0.581038; -0.039916; 0.000000; 0.000000; 0.000000								
5	2188298825124658; ACC; 0.638141; 3.853477; 9.684996; 0.000000; 0.000000; 0.000000								
6	2188298825975658; GYRO; 0.051138; -0.355524; -0.163380; 0.000000; 0.000000; 0.000000								
7	2188298844976221; ACC; 0.385572; 3.789993; 9.547080; 0.000000; 0.000000; 0.000000								
8	2188298845827221; GYRO; 0.020141; -0.221151; -0.244084; 0.000000; 0.000000; 0.000000								
9	2188298864829346; ACC; 0.108611; 3.776892; 9.492684; 0.000000; 0.000000; 0.000000								
10	2188298865680346; GYRO; -0.002635; -0.196158; -0.213541; 0.000000; 0.000000; 0.000000								
11	2188298884682471; ACC; 0.210556; 3.877793; 9.562978; 0.000000; 0.000000; 0.000000								
12	2188298885533471; GYRO; 0.010071; -0.193836; -0.100583; 0.000000; 0.000000; 0.000000								
13	2188298904534033; ACC; 0.376206; 3.991584; 9.463149; 0.000000; 0.000000; 0.000000								
14	2188298905385033; GYRO; 0.051784; -0.116623; -0.040614; 0.000000; 0.000000; 0.000000								
15	2188298924387158; ACC; 0.416246; 3.972804; 9.336343; 0.000000; 0.000000; 0.000000								
16	2188298925238158; GYRO; 0.091926; -0.114616; -0.050510; 0.000000; 0.000000; 0.000000								
17	2188298944240283; ACC; 0.326655; 4.101756; 8.905837; 0.000000; 0.000000; 0.000000								
18	2188298945091283; GYRO; 0.033353; -0.124634; -0.116309; 0.000000; 0.000000; 0.000000								
19	2188298964091846; ACC; 0.080349; 4.312877; 8.877260; 0.000000; 0.000000; 0.000000								

Fig. 2. CSV datasheet of Accelerometer and Gyroscope

We collected almost 28000 of 4 classes. The classes are- Stairup, Stairdown, Liftup, Lift down.

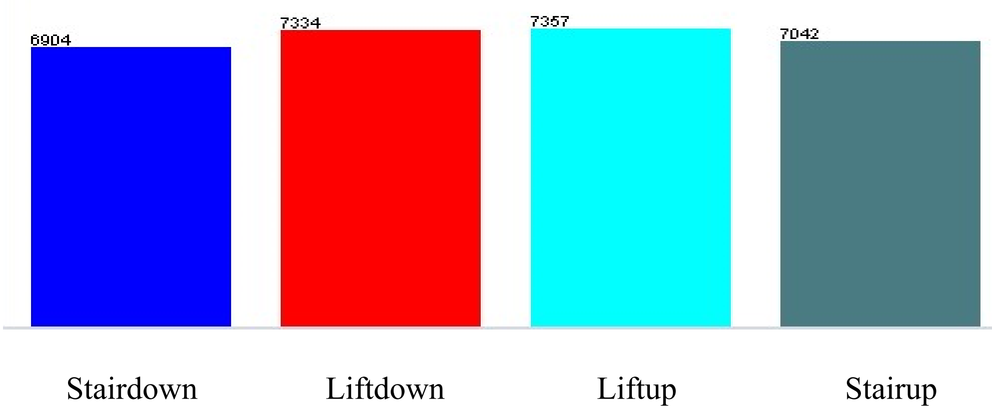

Fig. 3. The number of instances by class

We tried to maintain the weight of the classes almost the same to avoid data imbalance.

3.2 Data Preparation

To prepare the data, we had to merge the CSV files of each class separately. We split the data into columns and labeled them by class name to qualify it for classification.

	A	B	C	D	E	F	G
1	stairup	1.27E+15	ACC	2.260126	3.983952	10.6111	
2	stairup	1.27E+15	GYRO	0	0	0	
3	stairup	1.27E+15	ACC	2.834735	2.796427	8.044518	
4	stairup	1.27E+15	GYRO	-0.00304	-0.00929	-0.00456	
5	stairup	1.27E+15	ACC	3.562572	2.75812	8.121132	
6	stairup	1.27E+15	GYRO	-0.00808	-0.02382	-0.01264	
7	stairup	1.27E+15	ACC	3.409343	3.754108	9.653421	
8	stairup	1.27E+15	GYRO	-0.01019	-0.03044	-0.01766	
9	stairup	1.27E+15	ACC	2.528277	4.13718	10.68772	
10	stairup	1.27E+15	GYRO	-0.14338	-0.49084	-0.32535	
11	stairup	1.27E+15	ACC	1.302446	4.328717	11.60709	
12	stairup	1.27E+15	GYRO	-0.05063	-0.18396	-0.18412	
13	stairup	1.27E+15	ACC	0.881066	4.098873	12.52646	

Fig. 4. Labeled Datasheet

We used python and 'pandas' to access the CSV file and use the data as data frames.

3.3 Graph Analysis

As we collected raw data from the sensor, we plotted the data in the graph and analyzed it to see the difference. We noticed visible differences between each class. We used the 'matplotlib'

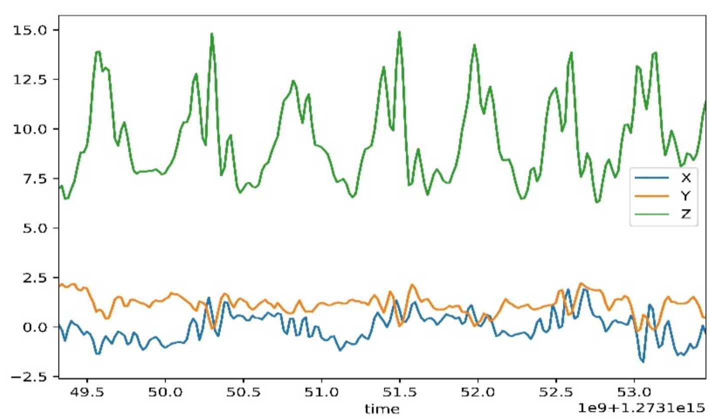

Fig. 5. Visualized Accelerometer data

library in python to visualize the data.

3.4 Classification

We used Support Vector Machine(SVM) to classify our dataset. The support vector machine algorithm finds a hyperplane in N-dimensional space that distinctly classifies the data points. There are many possible hyperplanes to separate the two classes of data points. We have to find a plane that has the maximum margin. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Hyperplanes are decision

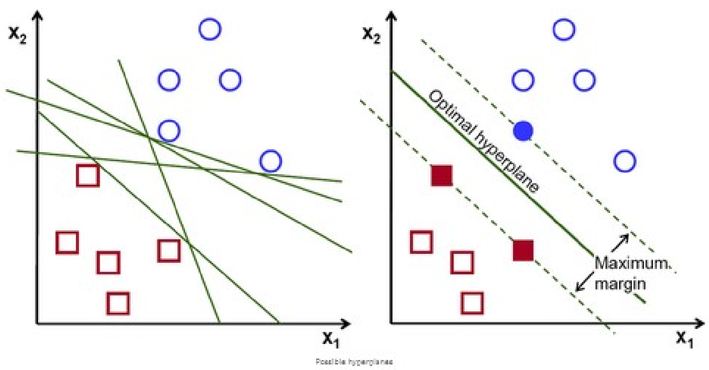

Fig. 6. SVM optimal hyperplane

boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes.[4]

We used the kernel trick as our data were not linearly separable. Kernal is a technique used to transform lower-dimensional features into higher dimensions features. It is a very important topic for classification, specifically, to classify nonlinear data to linear data. Given a set of training data

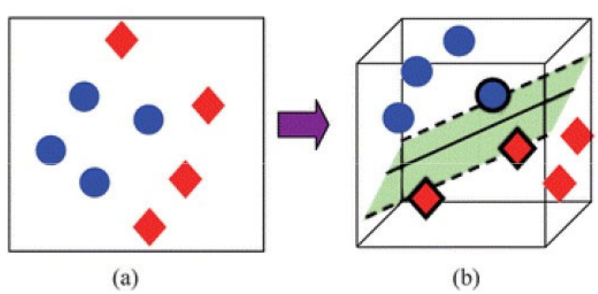

Fig. 7. Kernel Technique

that is not linearly separable; one can transform it into a training set that is linearly separable by mapping it into a possibly higher-dimensional space via some non-linear transformation.[5]

We used the RBF kernel to classify our dataset. We implemented it using Anaconda, Python, and the 'Sci-kit learn' library.

4. Results and Discussion

In this part, we discuss the results of our project. From the visualized graph of our dataset, we could see the differences in the sensor reading for different classes.

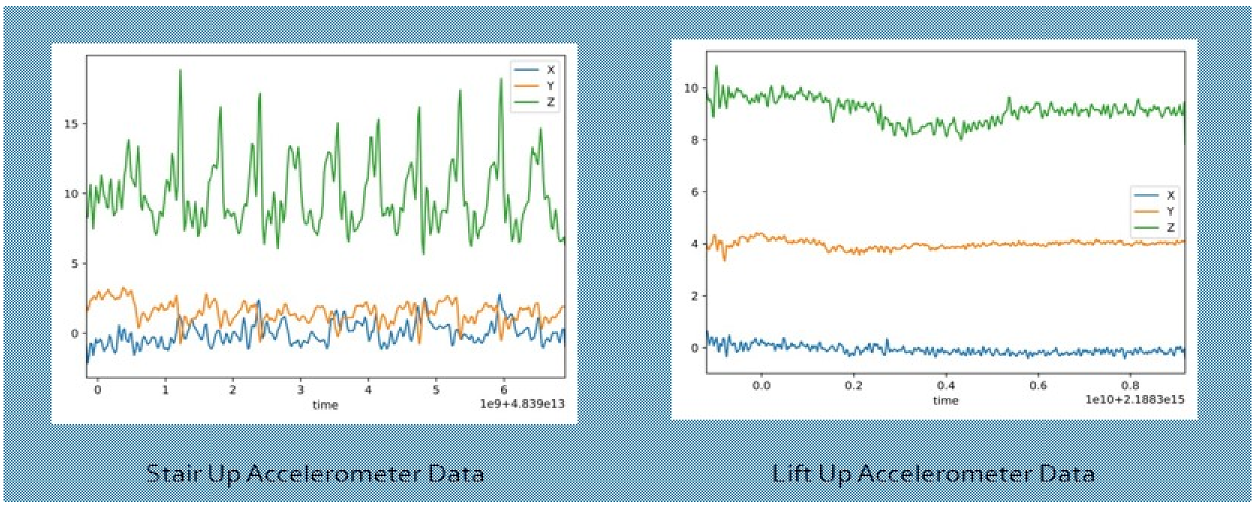

Fig. 8. Visualized Stair Up and Lift Up Accelerometer data

In the fig. 8, we can see there are visible differences between the staircase and lift accelerometer data.

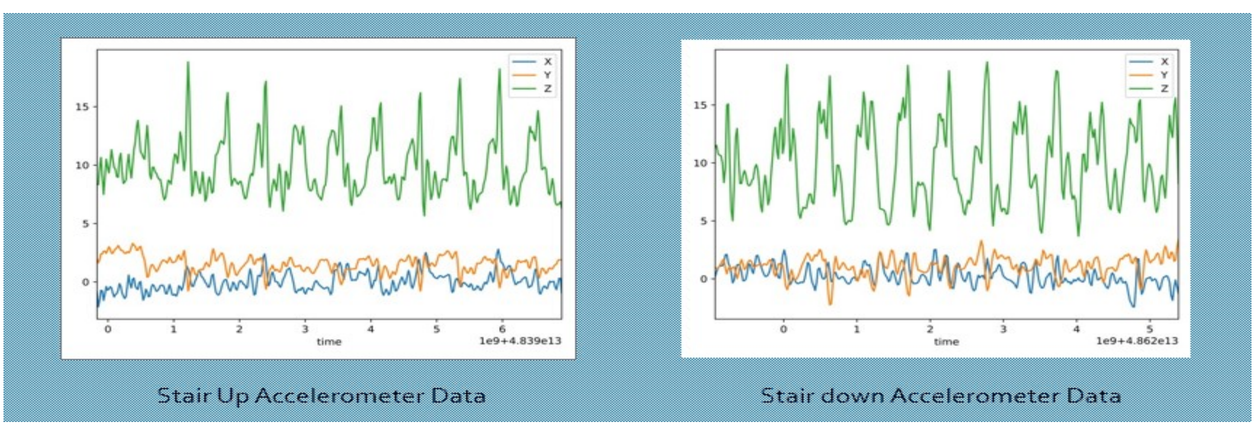

Fig. 9. Visualized Stair Up and Stair Down Accelerometer data

In this fig. 9, we can see there are also slight differences between the stair up and stair down accelerometer data.

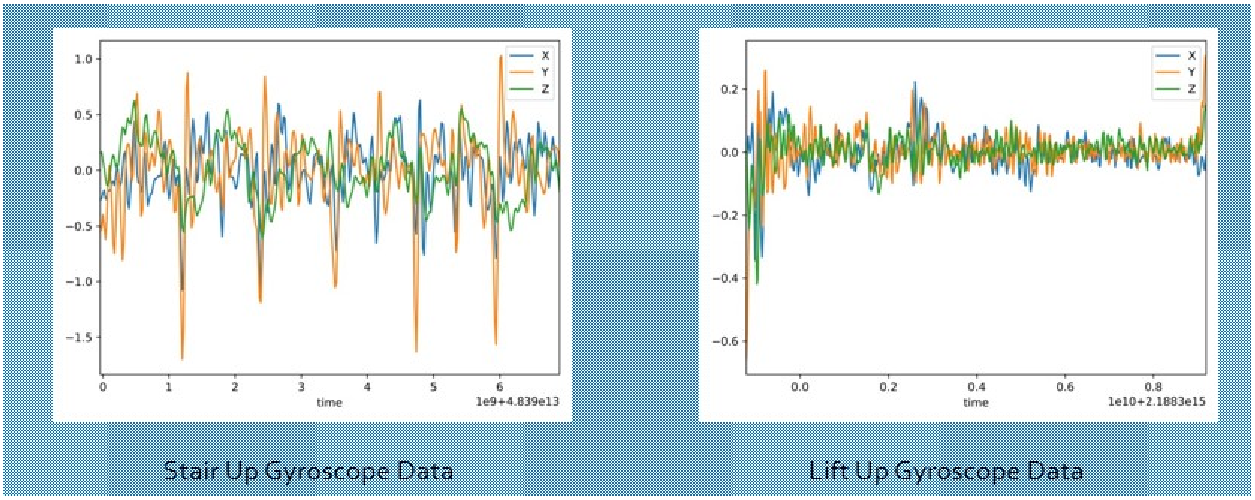

Fig. 10. Visualized Stair up and Lift Up Gyroscope data

In the fig .10, The differences in gyroscope sensor data can be seen. As it depends on orientation, it clearly depends on the data collection event.

We classified our data using Support Machine Vector(SVM). We used RBF kernel as our data is not linearly separable. At first, we shuffled our dataset. Then We split our dataset into a training set and test set by 80:20. We used C=50 and gamma=1 for the RBF kernel. The classification report is given below-

	precision	recall	f1-score	support
liftdown	0.82	0.87	0.84	1427
liftup	0.86	0.87	0.86	1513
stairdown	0.70	0.67	0.69	1402
stairup	0.70	0.69	0.69	1386
accuracy			0.78	5728
macro avg	0.77	0.77	0.77	5728
weighted avg	0.77	0.78	0.77	5728

Train accuracy: 0.7981142782312628
Test accuracy: 0.7760125698324022

Fig. 11. Classification report of SVM

From the classification report, we can see that we are getting almost 77.6% for our test set. Whereas, from our training set, we got 79.8% accuracy. We can say that out dataset fits the model. We ported our model to java to use it in developing android applications in the future.

5. Conclusions and Recommendations

In this project, we developed a model to differentiate between lift and staircase activity. We used smartphone sensors in our android device to collect our data and classified it using Support Vector Machine. We got almost 78% accuracy, which is suitable to differentiate between these two activities. Precision and accuracy can be increased by collecting data in a more controlled environment and by pre-processing the data to remove noise. We can also import the model in an android environment to predict new data from the sensor input.

6. References

[1] [Online] Gsmarena, “Sensors Definition,” https://www.gsmarena.com/glossary.php3?-term=sensors

[2] [Online] “Support Vector Machine, “ScienceDirect,” https://www.sciencedirect.com/topic-s/neuroscience/support-vector-machine

[3] Xing Su, Hanghang Tong, and Ping Ji, “Activity Recognition with Smartphone Sensors,” TSINGHUA SCIENCE AND TECHNOLOGY, June 2014

[4] [Online] TowardsDataScience, “Support Vector Machine — Introduction to Machine Learning Algorithms,” https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[5] [Online] Medium, “Kernel Trick and use in Support Vector Machines,” https://medium.com/kernel-trick/kernel-trick-and-use-in-support-vector-machines-ffb9d233a7fb