HUMAN ACTIVITY RECOGNITION USING SMARTPHONE SENSOR DATA

BACHELOR OF ENGINEERING

SEMESTER -VII

PROJECT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BIRLA INSTITUTE OF TECHNOLOGY

MESRA-835215, RANCHI

2020

Supervised By :- Mr. Narayan Sharma

Submitted by :-

GAURAV PRASAD	HARSH RAJ	SHUBHAM APOORVA CHANDARA
BE/10158/17	BE/10430/17	BE/10026/17

ACKNOWLEDGEMENT

In the present world of competition there is a race of existence in which those are having will to come forward and succeed. Project is like a bridge between theoretical and practical work.

We would like to express our special thanks of gratitude to our teacher Sir Narayan Sharma who gave us this golden opportunity to do this wonderful project of Human Activity Recognition.

The completion of the project could not have been accomplished without the support of the group members. Although this project has been prepared with utmost care and deep routed interest. Even then I accept respondent and imperfection.

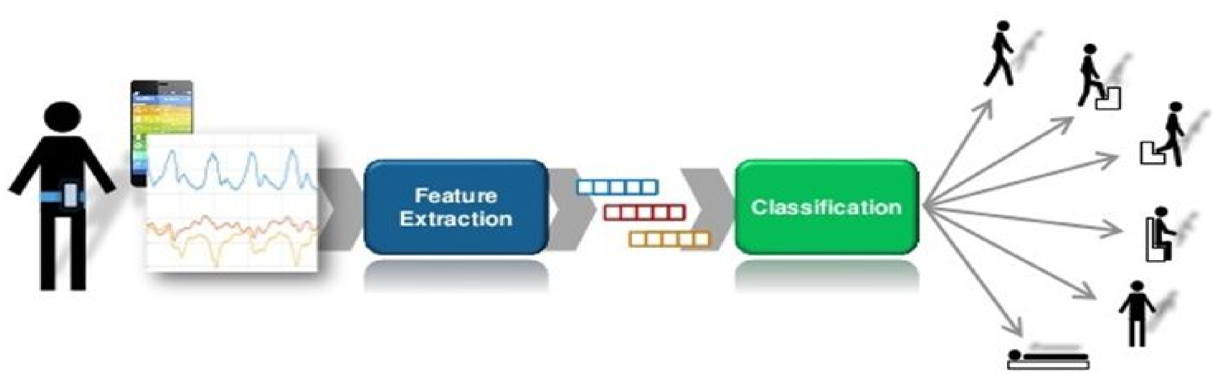
Contents

Project Objective

Using smartphone accelerometer data predict the physical activity being performed by its user. The activities being - walking, jogging, walking upstairs, walking downstairs, sitting, and standing.

Introduction

Today almost everybody has a smartphone and these smartphones are equipped with various sensors such as- GPS sensors, accelerometers, gyroscopes, and proximity sensors. The presence of such sensors in almost any affordable smartphone gives rise to enticing new opportunities for data mining and data mining applications. In this project we are going to use the readings of accelerometers to perform human activity recognition, which as the name suggests attempts to recognize the physical activity being performed by the user.



Human Activity Recognition Semantic Diagram

Proposed Method

The project can be divided into three major sections, the first section involves the transformation of the time series dataset, where we create segments of fixed time periods into which the dataset is split. We have chosen a ten seconds time period for each segment as it includes sufficient repetitions of an activity being performed.

In the second section, we use the transformed dataset to train our neural network model. The dataset contains accelerometer readings for x, y, and z directions, these are our three features using which we train the Long Short-Term Memory (LSTM) neural network which is a modified form of recurrent neural networks. After training we export the model to be used in the android application.

In the third section, we use the exported tensorflow model or the tensorflow graph in our android application which uses live accelerometer readings as prediction input to determine the physical activity being performed by the user.

About The Dataset -

We will be working with android based smartphones and have chosen the Wireless Sensor Data Mining(WISDM) data set. The data has been collected through controlled laboratory experiments. The dataset contains over a million observations for 36 users for 6 activities combined. The six activities are - walking, jogging, walking upstairs, walking downstairs, standing, and sitting.

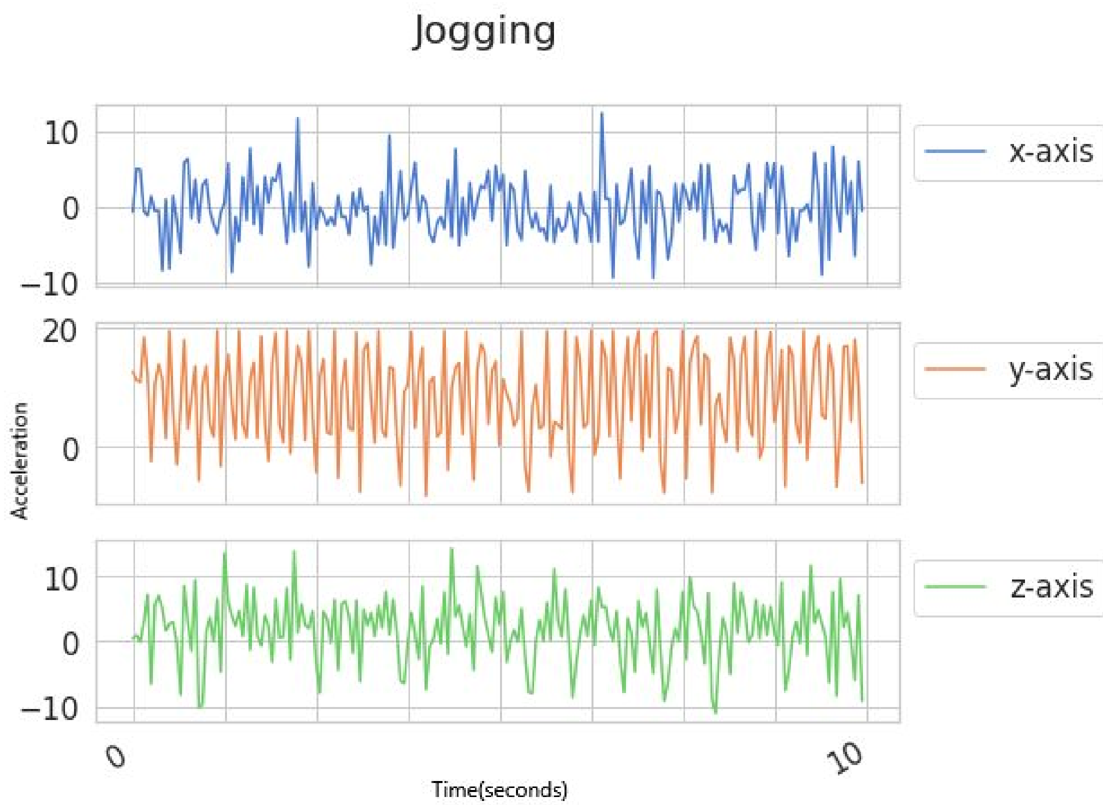
In the experiments used for collecting the accelerometer data, a single device - a modern smartphone, is used rather than multiple accelerometer devices placed at various positions on the user's body. No further equipment or actions are needed to be performed by the user.

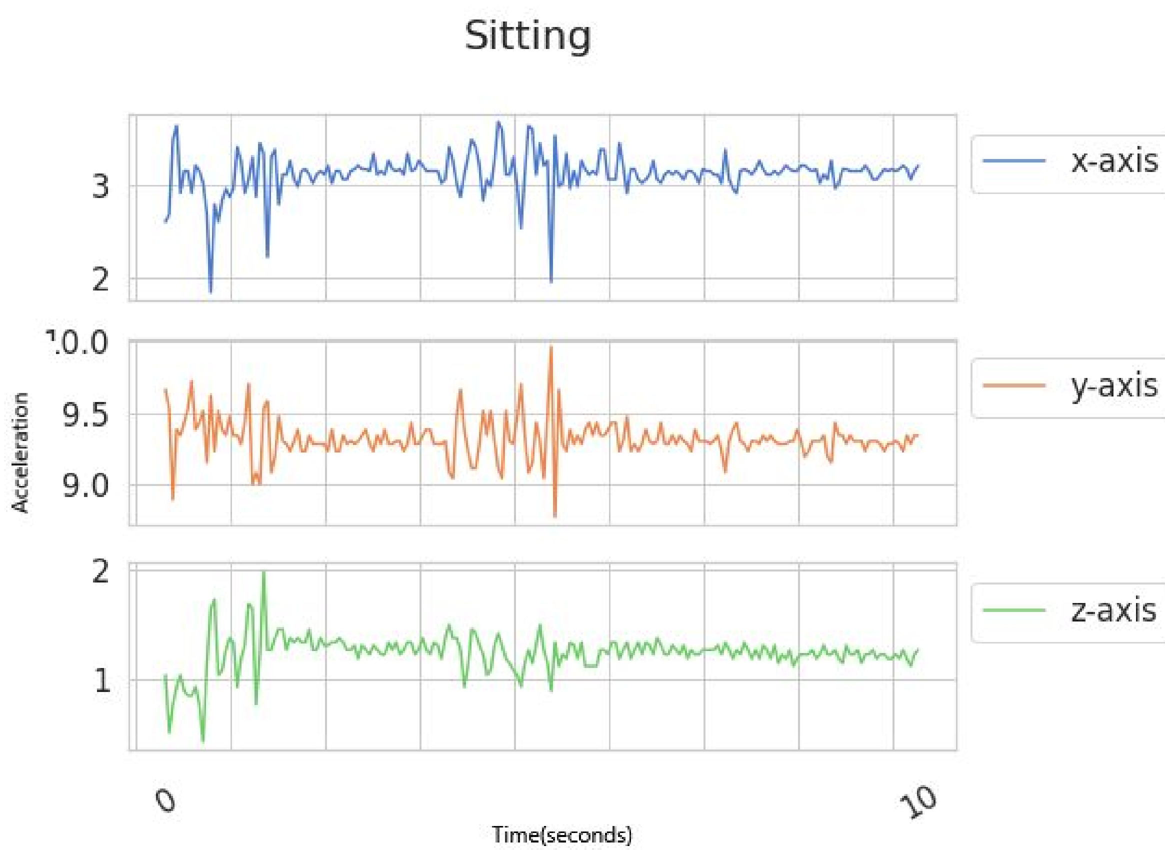
The dataset contains six columns - the first three columns are used for user-id, activity label, and timestamp; the last three columns are for the accelerometer readings for x,y, and z directions respectively. The accelerometer data has been collected at a frequency of 20Hz, therefore, we have 20 observations per second.

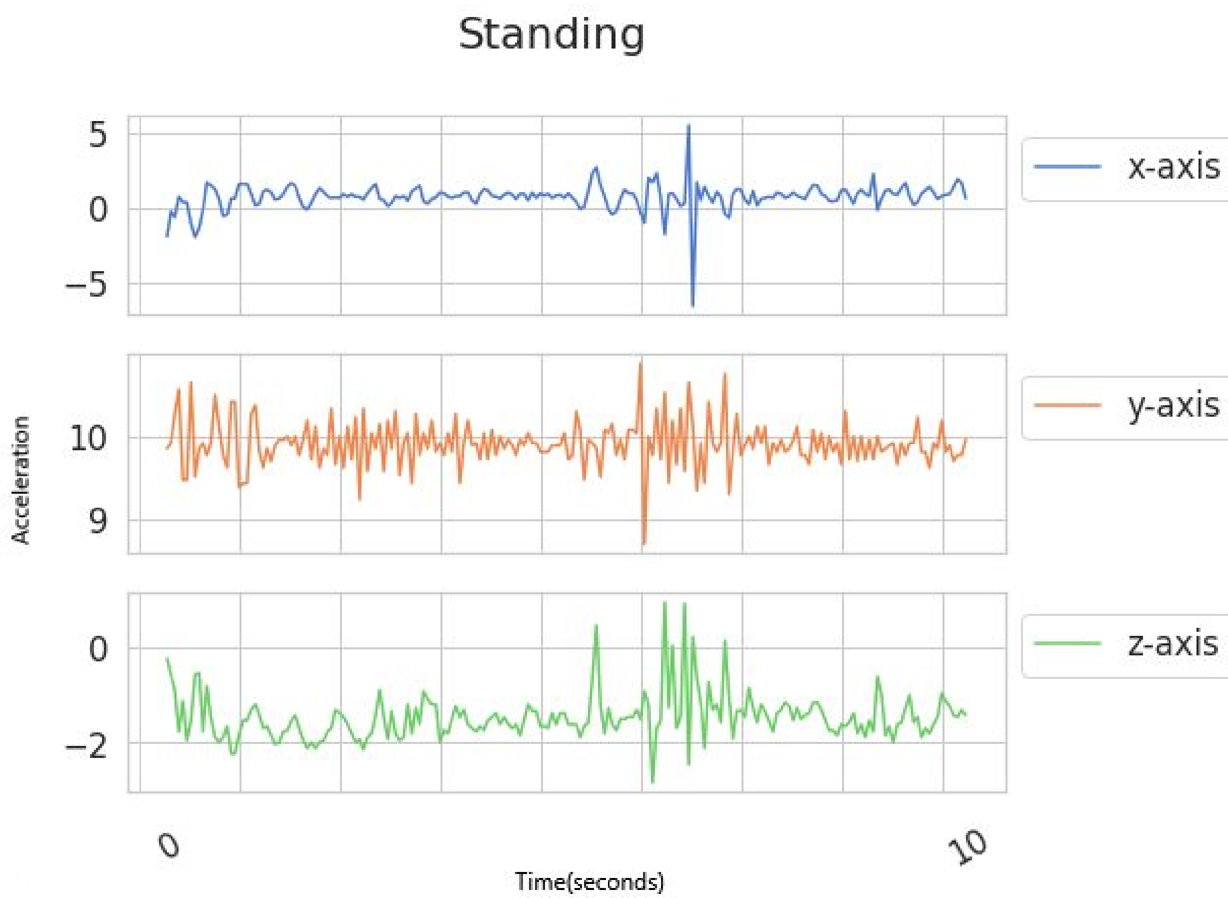
	user	activity	timestamp	x-axis	y-axis	z-axis
0	33	Jogging	49105962326000	-0.694638	12.680544	0.503953
1	33	Jogging	49106062271000	5.012288	11.264028	0.953424
2	33	Jogging	49106112167000	4.903325	10.882658	-0.081722
3	33	Jogging	49106222305000	-0.612916	18.496431	3.023717
4	33	Jogging	49106332290000	-1.184970	12.108489	7.205164

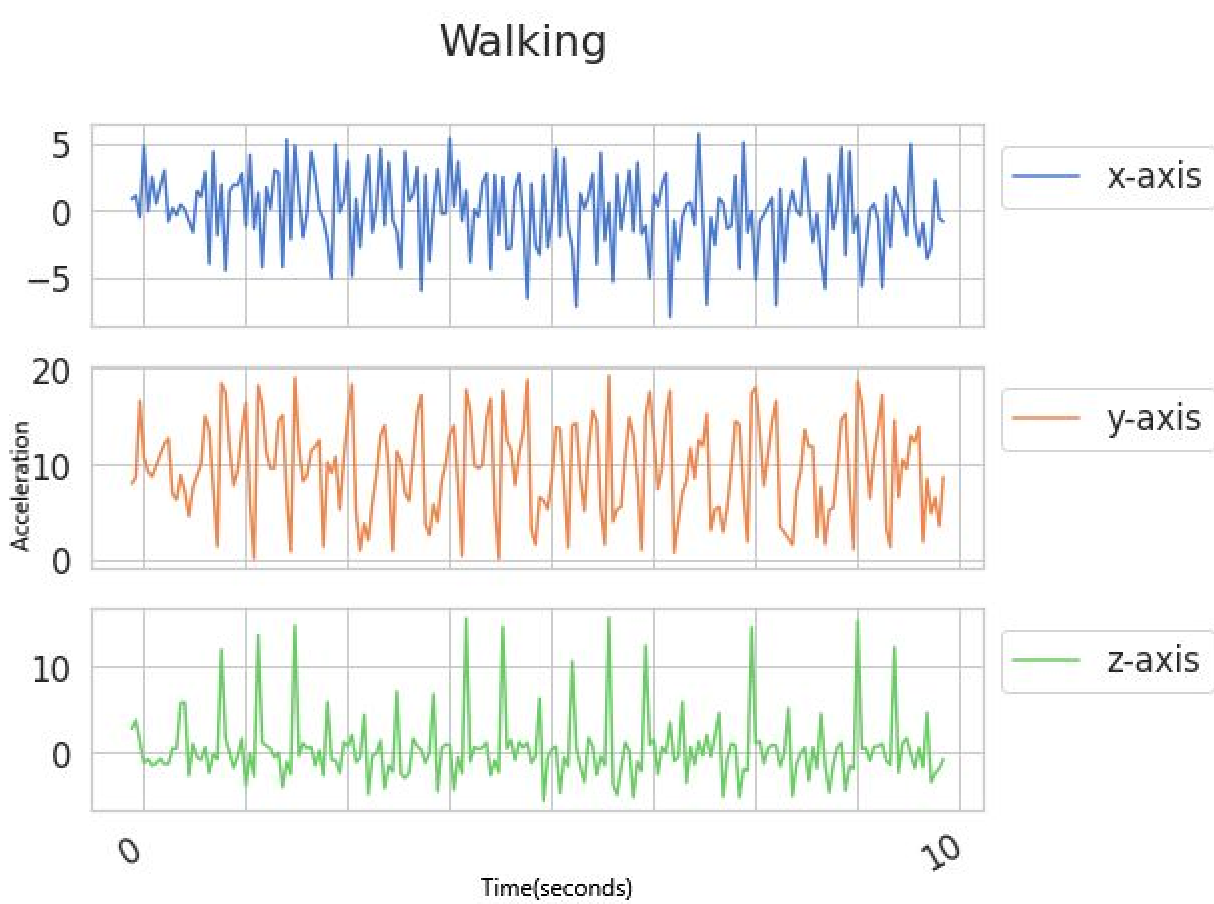
Dataset Head

Let's take a look at the accelerometer readings for some of the activities for 10 second time intervals or 200 readings -






Standing
x-axis
y-axis
z-axis
Acceleration
Time(seconds)
5
0
−5
10
9
0
−2
0
10


Walking
x-axis
y-axis
z-axis
Acceleration
Time(seconds)
5
0
−5
20
10
0
10
0
0
10

Data Preprocessing - Accelerometer readings are being recorded at a frequency of 20Hz, i.e. we have 20 readings for every second. The raw data is available in the form of time series data, we shall generate observations by transforming the data into 10 second segments. The number of readings taken in a time interval of 10 seconds is 200, therefore, each segment contains 200 readings.

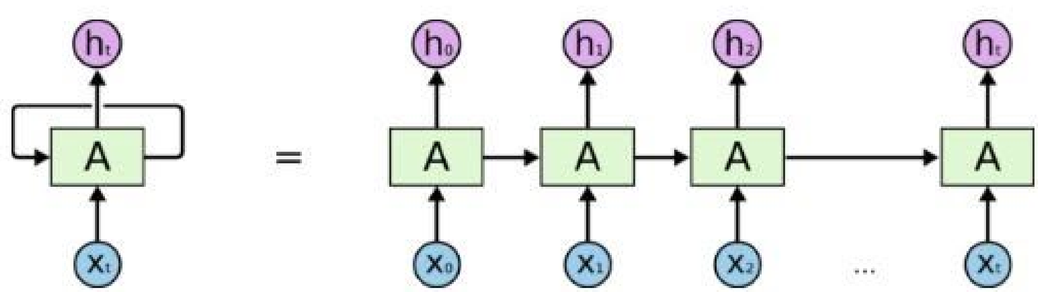
The 10 seconds time interval provides sufficient time to record multiple repetitions of the motion involved in the six activities. The number of features is 3 which comprises the accelerometer readings in x,y, and z directions.

The dataset has been split into training and testing sets in 80:20 ratio respectively.

Building The Model - Classical machine learning approaches like support vector machines and random forests require extensive feature engineering, which require in depth domain knowledge. In the [UCI HAR research]() on human activity recognition using smartphone sensor data, feature creation is one of the largest and most important tasks, over 500 features are created for the classical models to be able to predict the activity being performed with acceptable accuracy.

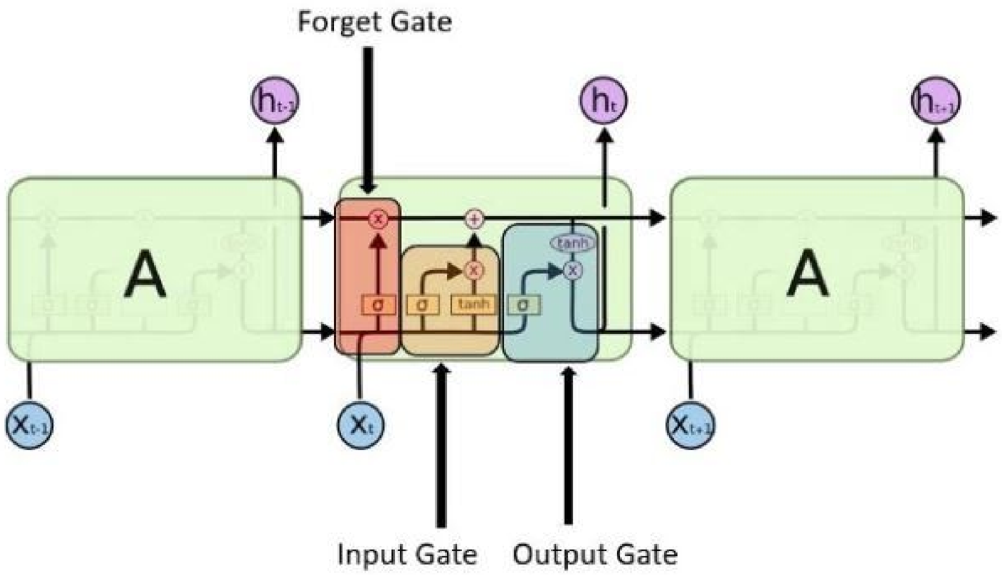
Recurrent Neural Networks (RNN), especially Long Short Term Memory (LSTM) neural networks which are a special kind of RNN are able to predict the activities accurately with little to no feature engineering.

LSTM networks are a modified form of RNNs that are able to easily remember past data in memory. LSTMs are able to process, classify and predict time series, which makes them suitable for training on our time series dataset with ten second intervals.



An unrolled recurrent neural network.

We have a sequence of inputs from X_0 to X_t, and h_0 to h_t serve as the input to the next layer. First it takes X_0 as the input and produces h_0 as the output. X_1 along with h_0 is the input for the next step. This keeps going on for all the nodes in the specific layer, this is how it keeps remembering the context during training.



LSTM Gates

- **Input Gate -** Decides which value from the input is to be used to modify the memory. It is the sigmoid function that decides upon the values to let through and the tanh function assigns weights to the passed values to determine their importance from -1 to 1.
- **Forget Gate -** Decides which details are to be discarded from the block, the decision is made through the sigmoid function. The previous state h_{t-1} and current input X_t are used to make the decision.
- **Output Gate -** The block's memory and the input are used to decide the output. The values to let through are decided by the sigmoid function and the tanh function assigns weightage to the values that decide their level of importance from -1 to 1.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Input gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Forget gate

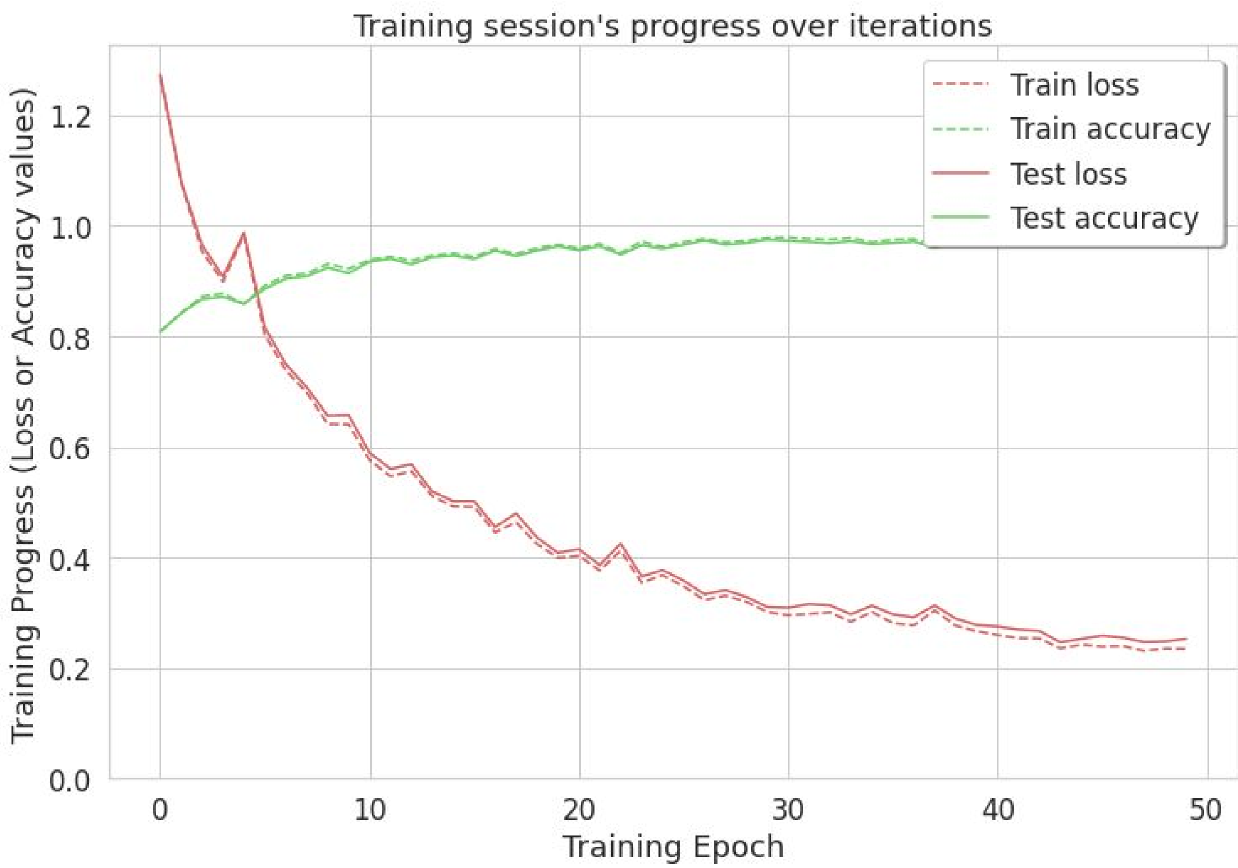
$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Output gate

Our model consists of 2 LSTM layers, the first LSTM layer is BasicLSTMCell, which has 64 hidden units - the number of units in the LSTM Cell. The second RNN layer takes in as input the output of the previous LSTM layer, i.e. the list of the RNN cells that will be composed in this order.

For our LSTM model we are using the Adam optimization algorithm which is an extension of the stochastic gradient descent. Tensorflow provides us with the AdamOptimizer function which takes in the learning rate and exponential decay rates its parameters. We have set the number of epochs to 50, and the batch size to 1024. After training is complete the model is saved to disk and exported to be used in the android application.

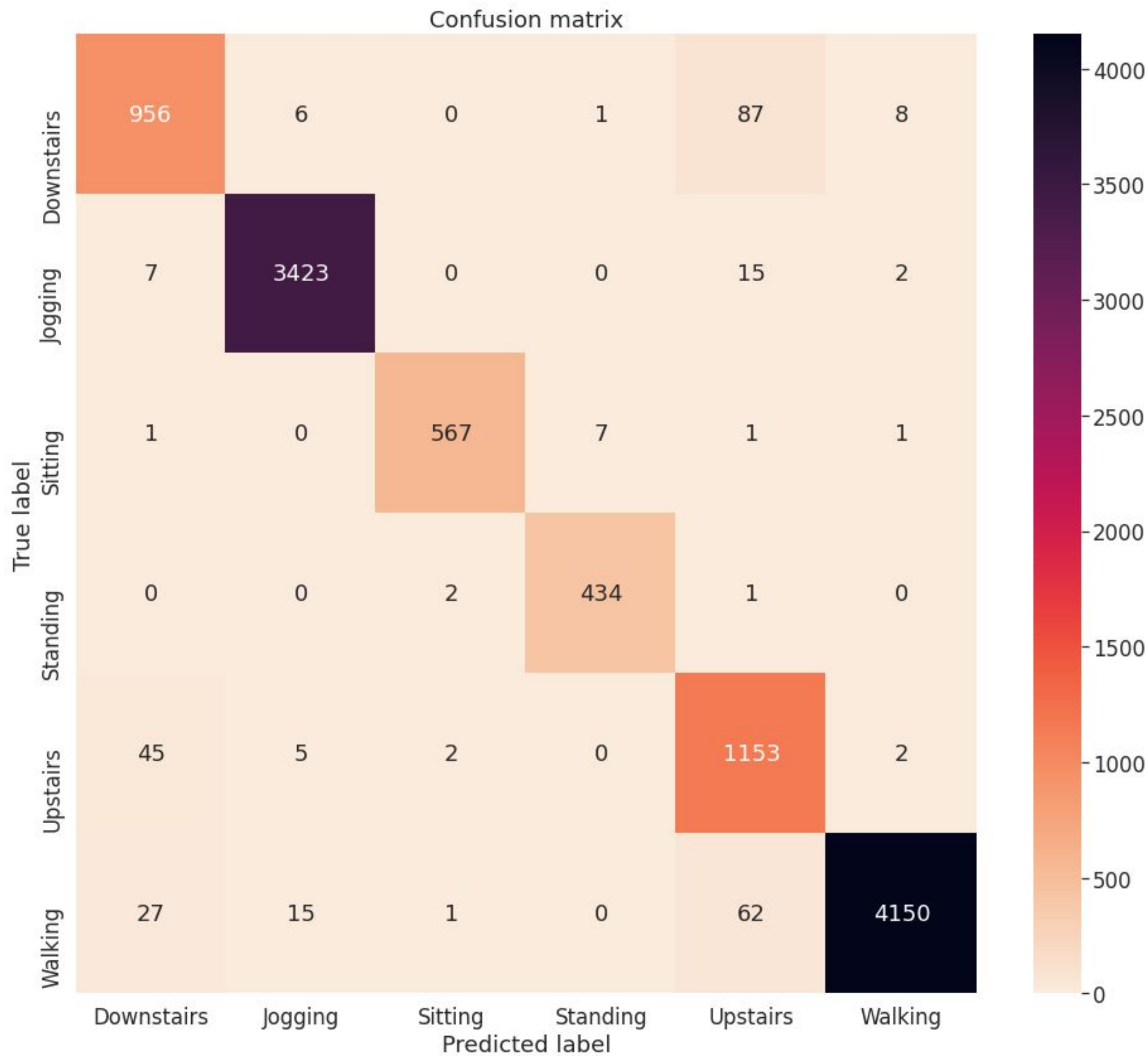
Model Evaluation - Let us take a look at the accuracy and loss as the model trains for more number of epochs. Finally we shall also create a confusion matrix to gain more insights.



Training Session Progress

Throughout the training process test set accuracy has gone up to 97% and the test loss has gone down to 0.25.

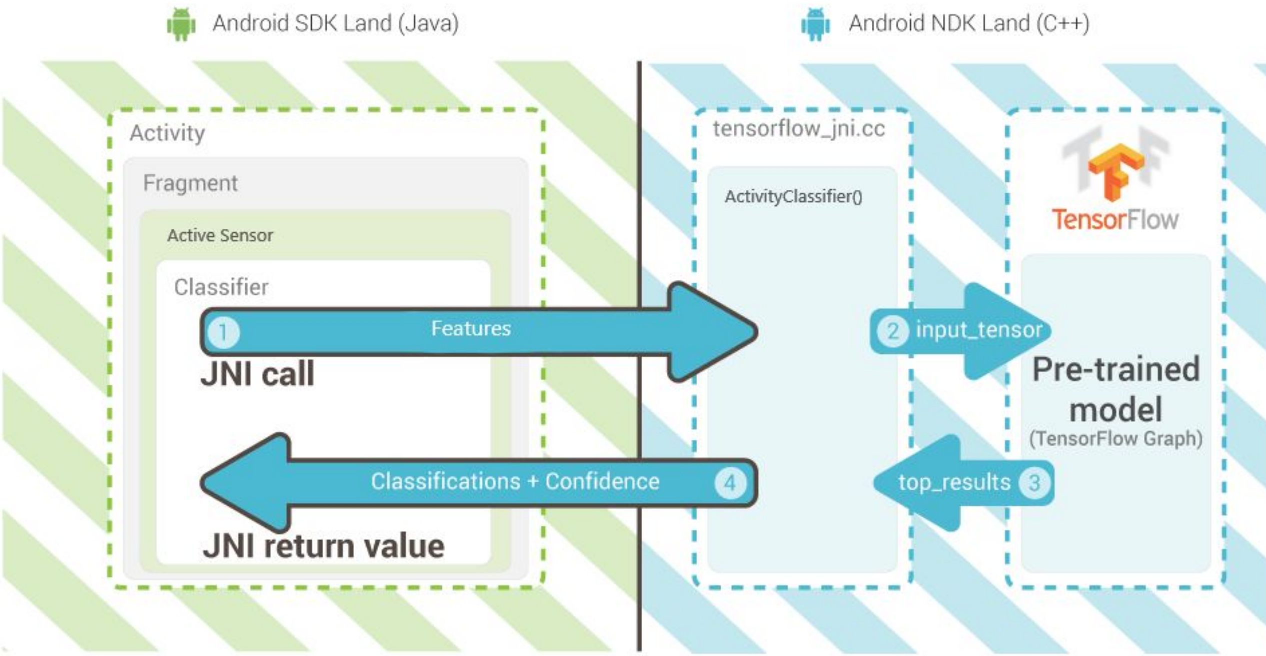
The Confusion Matrix -



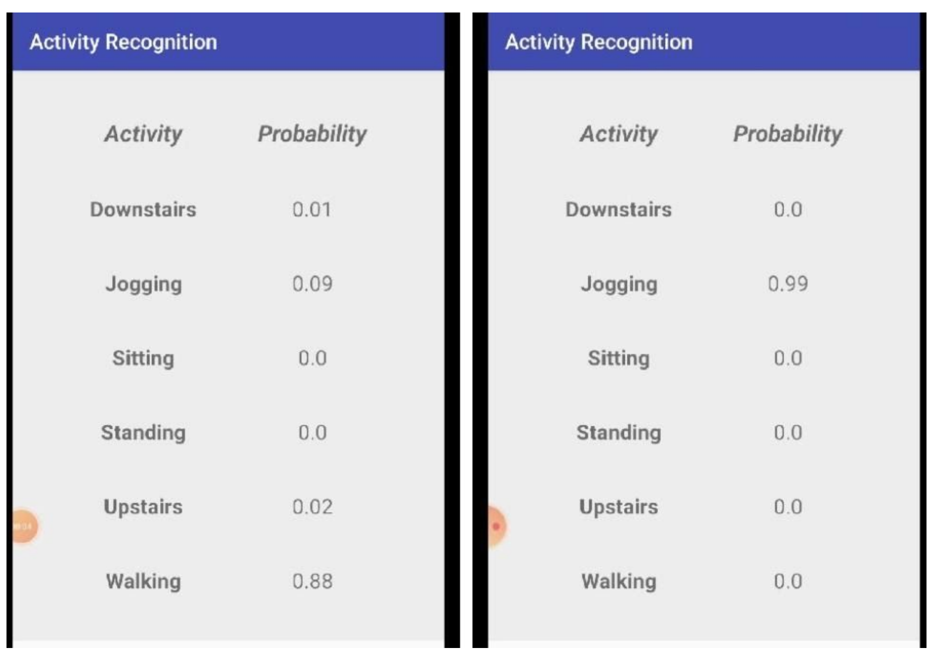
Confusion Matrix

We have achieved an accuracy of 97.28% on the test data.

Android Application Workflow -



The Android application uses the exported tensorflow model to predict the activity being performed. The application takes in the log data of the accelerometer (x, y, and z directions) as its input/features, the sensor input is converted to tensors every 10 seconds just like we did in the data preprocessing stage. The input tensor is then given to the pre-trained model for prediction. The model returns probabilities associated with the respective activities to the application which is then displayed on the screen for the user to see.



Working of the Android Application.

CONCLUSIONS AND FUTURE SCOPE:

We described how a device like a smartphone can be used to perform activity recognition, while it is kept in one's pocket. Our results further showed that activity recognition can be very accurate, with most of the activities being recognized with a success rate of over 90%. In addition, these activities can be recognized without much delay, since each instance is generated with only 10 seconds worth of data.

In the future we intend to improve our activity recognition application by enabling it to recognize more activities like cycling and driving, we can also improve the application by making it so that it is able to recognize activities when the smartphone or other wearable device is at some other position like the wrist or upper arm. We thought of several interesting applications for human activity recognition.

Some of these applications include :

- *Activity recognition can be very useful in the fitness industry where wearable devices are generating tons of sensor data , using these one can provide the user with accurate real time information about their state.*

- *This model can be used to design a fall detection software for the elderly people which will reduce the time in taking appropriate measures if it happens.*

- *Customization of the smartphone based on the general activity trends of the user.*

We believe that mobile sensor data provides tremendous opportunities for data mining and we intend to leverage our Android-based data collection/data mining platform to the fullest extent possible.

References

1. Activity Recognition using Cell Phone Accelerometers, Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore, SensorKDD (2010).
2. Real-time human activity recognition from accelerometer data using Convolutional Neural Networks, Andrey Ignatov, ETHZ, Switzerland (2017).
3. Recurrent Neural Networks and LSTM explained, Purnasai Gudikandula (2019).
4. A Review of Human Activity Recognition Methods, Front. Robot. AI. Michalis Vrigkas , Christophoros Nikou, Loannis A Kakadiaris (2015).
5. Wireless Sensor Data Mining (WISDM) Dataset, (2012).
6. TensorFlow-Version 1.15 Documentation .
7. Android Studio-Version 4.1 Documentation.