Human activity recognition on smartphones

# Overview

## Introduction

|  |  |
| --- | --- |
|  | A variety of real time sensing applications are becoming available, especially in the life logging, fitness domains. These applications use mobile sensors embedded in smartphones to recognize human activities in order to get a better understanding of human behavior. HAR system is required to recognize six basic human activities such as walking, jogging, moving upstairs, downstairs, running, sleeping by training a supervised learning model and displaying activities result as per input received from our accelerometer sensor and CNN model. |

## Project Background and Description

|  |  |
| --- | --- |
|  | The first HAR approach contains a large number of sensor type technologies that can be worn on-body known as wearable sensors, ambient sensors, and, together, both will make hybrid sensors that help in measuring quantities of human body motion. Various opportunities can be provided by these sensor technologies which can improve the robustness of the data through which human activities can be detected and also provide the services based on sensed information from real-time environments, such as cyber-physical-social systems there is also a type of magnetic sensors when embedded in smartphone can track the positioning without any extra cost. 2. Vision-based—RGB video and depth cameras being used to obtain human actions. 3. Multimodal—Sensor’s data and visual data are being used to detect human activities |

## Project Scope

|  |  |
| --- | --- |
|  | Project scope defines the boundaries of a project. Think of the scope as an imaginary box that will enclose all the project elements/activities. It not only defines what you are doing (what goes into the box), but it sets limits for what will not be done as part of the project (what doesn’t fit in the box). Scope answers questions including what will be done, what won’t be done, and what the result will look like. |
|  |  |

## Project Proposal and Plan

|  |  |
| --- | --- |
|  | Idea for this project is to collect data then some preprocessing is to be done on raw collected data. After balancing and standardizing it will be plotted on scatter plot by using matplot library. Then by using these graphs frame preparation is to be done. After that CNN model will be used to classify human activities. Then for accuracy measurement learning curve and confusion matrix will be plotted. |

|  |  |  |
| --- | --- | --- |
| **Week** | **Goal** | **Comments** |
| Week 4 | Project Proposal | Project proposal presentation |
| Week 5 | Data Collection | Collection of sensor based WISDM dataset |
| Week 6-8 | frame preparation | Visualization and frame preparation |
| Week 8 | model implementation | CNN, RNN-LSTM models implementation |
|  |  |  |
| Week 9-11 | Transition recognition | Different type of transition of activities recognition by LSTM |
| Week 12-13 | Submission | Final presentation and report submission |

## Data set

|  |  |
| --- | --- |
|  | Human-centered computing is an emerging research field that aims to understand human behavior and integrate users and their social context with computer systems. One of the most recent, challenging and appealing applications in this framework consists in sensing human body motion using smartphones to gather context information about people actions. In this context,The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The objective is to classify activities into one of the six activities performed.  The trials were conducted on a group of 30 participants ranging in age from 19 to 48. Each participant used a smartphone (Samsung Galaxy S II) while doing six activities (walking, climbing stairs, walking down stairs, sitting, standing, and lying). We recorded 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using its integrated accelerometer and gyroscope. The tests were videotaped so that the data could be manually labeled. The resulting dataset was divided into two sets at random, with 30% of the participants chosen to create test data and 70% of the participants chosen to create training data. |
|  | The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butter worth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain. |

## Attribute Information

|  |  |
| --- | --- |
|  | Attribute Information For each record in the dataset the following is provided:   * Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration. * Triaxial Angular velocity from the gyroscope. * A 561-feature vector with time and frequency domain variables. * Its activity label. * Features are normalized and bounded within [-1,1]. * An identifier of the subject who carried out the experiment |

## Applied Machine learning methods

|  |  |
| --- | --- |
|  | Six machine learning methods were tested with default settings. These were all part of the Python Sci-kit Learn package [6], namely: Decision Tree Classifier, K-Neighbors Classifier, Support Vector Classifier, Gaussian Naïve Bayes Classifier, Quadratic Discriminant Analysis Classifier, Multi layer Perceptron Classifier. As far as the target variable was discreet and categorical, accuracy (=number of correct predictions / numbers of all predictions) was used as a measurement of the goodness of the models. |

|  |  |  |
| --- | --- | --- |
| **Rank** | **Machine learning model** | **Accuracy** |
| 1 | MultiLayer Perceptron Classifier | 0.94095 |
| 2 | Support Vector Classifier | 0.93077 |
| 3 | Decision tree classifier | 0.86155 |
| 4 | K-Nearest Neighbour classifier | 0.80726 |
| 5 | Gaussian naive baysed classifier | 0.77207 |
| 6 | Quadratic discriminant Analysis Classifier | 0.73566 |

|  |  |
| --- | --- |
|  | The results of the six machine learning methods mentioned above were compared by their accuracy using the same training set. The results were sorted by the accuracy in Table II. Based on the values represented in the table the Multilayer Perceptron Classifier reached the highest accuracy on the training set, while Support Vector Classifier also showed a reasonable accuracy value. Therefore, this study tries to find the best parameter settings for these two Classifiers. |

## Libraries that I have used in my project

|  |  |
| --- | --- |
|  | 1.Pandas for loading dataset  2.Numpy for performing numerical computation  3.Matplotlib for plotting  4.Pickle to serialize the object for permanent storage  5.Scipy for different scintific computation and statistical functions  6.Tensorflow for creating different neural networks  7.Seaborn for beautifying graphs  8.Sklearn for training and testing splitting of data and for the metrics that I will be using to judge my model. |

# Approval and Authority to Proceed

We approve the project as described above, and authorize the team to proceed.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| **Approved By** |  |  | Date |  | Approved By |  |  | Date |