Human Gait Activity Recognition

1. Abstract
2. Methods
   1. Phase 1: Preprocessing
   2. Phase 2: Feature Extraction
   3. Phase 3: Classification Techniques
3. Results and Conclusion
4. Challenges
5. **Abstract**

As we now Activity Recognition is useful in many fields as pervasive healthcare and surveillance, The recognition of human gait can be useful to identify the characteristics of the places or physical space such as walking , walking downstairs …etc.

In this Project a method of recognizing gait activities using acceleration data obtained from smartphone, the acceleration signal were segmented based on the strides (Peaks and Valley ) then Feature vector of the segmented signals was extracted, Which was used to train five Classifier Using:

-Support Vector Machine

- Decision Tree

- Random Forest

-Naive Bayes

- K-Nearest neighbors

Our Dataset:

It was collected from data consist of five Activities [ stand, walking, stair up, stair down, sit]

Where each activity has its own accelerometer data and time stamps

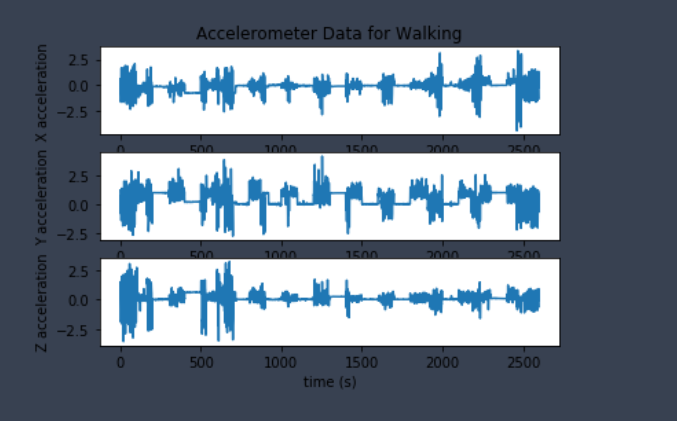
1. **Method**

Our Method are Classified into 3 Phases

* 1. **Phase 1: Preprocessing**
     1. **Load Data**

First thing we load our data, then we find that its consists of 5 columns “ time Elapsed , x, y, z, label “ and it consists of 2600 row

We plot Axes of Accelerometer Data:



* + 1. **Appling Low\_ pass Filter**

A low-pass filter is applied to accX, accY, accZ, we use the coefficient a which filter the highest frequencies with **value 0.5** the plot after filtration was :

**A screenshot of a computer

Description automatically generated with low confidence**

* + 1. **Applying smoothing**

By using of gaussian filter1d from SciPy library with sigma value = 4 we smoothed the accelerometer axes data and the plot :

A picture containing chart

Description automatically generated

* + 1. **Applying XYZ**

Now we try to find magnitude of the three axes and w make this on smoothed and filtered x, y, z acceleration data by applying

Plot for Data after applying:

A screen shot of a computer

Description automatically generated with low confidence

Graphical user interface, chart, histogram

Description automatically generated

* + 1. **Detection of Peaks and Valleys**

We use find\_peaks function from SciPy library we use it also for detect valleys by multiplying XYZ\_smoothed values with -1 and the peaks

A screenshot of a computer

Description automatically generated with medium confidence

* + 1. **Segmentation**

Applying segmentation depends on valleys of dataset XYZ\_Smoothed then we apply this segmentation on XYZ\_magnitude:

A screenshot of a computer

Description automatically generated with medium confidence

* 1. **Phase 2: Feature extraction**
     1. **Peak Height**
     2. **Mean**
     3. **Peak width**
     4. **Standard Division**
     5. **Data Frame for feature**

After we extract features we put this feature in data frame (df\_feature) then we labeled each features row according to actual label

for example : as we know the from time 0 to 99 there is activity done which is waking ,the first 12 valley points are from 0 to 90 so they are labeled as Walking and so on

we found feature data consists of 151 row and 5 columns

* 1. **Phase 3: Classification**
     1. **Train\_Test\_Split**

We split Feature data set to X And Y where

X is (acc\_mean, acc\_Std, peak\_height, peak\_width)

Y is labels

Then we use function train test split from scikit-learn library and split data with 0.1 for test and random state = 49

* + 1. **Preprocessing for Features data**

we use here preprocessing label encoder to convert Y to label and Standard Scaler to make feature scaling to X data

* + 1. **Algorithms Used**

- Support Vector machine:

We use Kernel ‘rbf’ , random state =0 and fit with train data and predict with test

* K-Nearest Neighbors:

We use n\_neighbor =3 metric= 'minkowski’ p=3

* Random forest:

n\_estimators = 17, criterion = 'entropy', random\_state = 0

- Decision Tree: max\_depth = 5

- Naive Bayes:

1. **Results and conclusions:**

To Calculate accuracy we use accuracy score function from metrics in sklearn

|  |  |
| --- | --- |
| Algorithm | Accuracy |
| SVC | 0.6875 |
| Decision Tree | 0.6875 |
| Random forest | 0.625 |
| KNN | 0.5625 |
| Naive Bayes | 0.5625 |

1. **Challenges:**

We Face a lot of challenges as paper ofRecognition of Gait Activities Using Acceleration Data from A Smartphone and A Wearable Device. Has many function with no identification for it like Power( ) , Forward Direction , segmenting and power for feature extraction

Also dataset we didn’t find perfect dataset and we collect dataset manually from different accelerometer read for activity and it was so small