**Road issue recognition using accelerometer and gyroscope data**

The goal of this project is to classify the road issues experienced by cycling (potholes, bumps, concrete expansion joints, and storm drain basins) from the 3D accelerometer and gyroscope data.

**Potential Technology to use**

Signal processing: Python, Numpy, Scipy and Matplotlib Classifier design: Tensorflow-Keras

**Input Data Reading**

Input data is read from .csv files provided.

*Data structure of the collected data*

**timeRelativeRef**(ms), **samplingTime** (ms),

**epoch** (s)**, latitude** (decimal degrees) **, longitude** (decimal degrees)**, altitude** (decimal degrees)**, speed** (m/s^2)**, satellites** (int)**,**

**accelerationX** (m/s^2), **accelerationY** (m/s^2), **accelerationZ** (m/s^2),

**roll** (degrees per second), **pitch** (degrees per second), **yaw** (degrees per second),

**temperature, humidity, barometric pressure, light,**

**PM1.0 concentration** (CF=1,Standard particulate ug/m3),

**PM2.5 concentration** (CF=1,Standard particulate ug/m3),

**PM10.0 concentration** (CF=1,Standard particulate ug/m3),

**PM1.0 concentration** (Atmospheric environment ug/m3),

**PM2.5 concentration** (Atmospheric environment ug/m3),

**PM10.0 concentration** (Atmospheric environment ug/m3)

*Example Row Data****WITHOUT GPS LOCATION***

*11924064,11924064,,,,,,,-0.18,-0.02,1.00,1.71,-5.98,-2.93,,,,,,,,,,,*

***WITH GPS LOCATION (only 1 sample every second)***

| *11925063,39,1649193000,55.6815262,12.5762043,10.6999998,0.87,8,-0.27,-0.06,0.98,1.59,-5.80,-2.99,26.53,29.69,99.88,255,5,7,7,5,7,7,* |
| --- |

**Preprocessing**

Raw data plots in time domain:

Signal we have been provided can contain noisy frequency components. Thus it is necessary to pass the signal through necessary filters and refine the data. Following plots show the raw training signal data in the time domain. Plots are X, Y, Z acceleration amplitudes of potholes, bumps, concrete expansion joints, and storm drain basins respectively. No conclusions can be made by just looking at the training data in the time domain. Thus to analyze the frequency content of the signal wrt time, a spectrogram function should be created.

**FFT Plots**

FFT plots give us a rough idea of the frequency content in the signal. Since, this is a time varying frequency signal, just looking at the FFT plots gives us no information about the time dependencies frequencies i.e how frequency content varies with respect to the time. This information can be obtained by spectral analysis.

**Spectrogram analysis**

In a spectrogram, the time varying signal is divided into a number of blocks along the time axis. These blocks are also known as ‘windows’. And then, SFFT is applied on each of these blocks. Resulting in the graph, that gives us the frequency distribution at different time steps. There are many window functions that can be used for this purpose. Following plots show the different spectrograms obtained by using different windows.

**Filter Design**

**Classifier Design**

There are multiple traditional ML classifier options available for this problem. For eg: SVM(State vector machines), Decision Trees, KNN(K-Nearest Neighbors), etc.

**Data Preparation**

First training and testing data can be split into 70% and 30%. Both the data should pass through the bandpass filter, to get rid of the noisy signals. Both training and testing data should be normalized. Because, we have three features (x, y, z) with different scales. Normalizing data accelerates the learning process.

**Data Reshaping**  
Regarding data reshaping consider that the sensor is reading the data (gyroscope and accelerometer) at the rate of 104 Hz, GPS data and air quality at a rate of 1 Hz.