ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

LOW-COST SENSOR FOR SEAMLESS ROAD QUALITY MONITORING

By

Team No.11

Yichen Li

Yihang Yang

Yuhang Chen

Zhengwei Dai

TA: Xiaoyue Li

May 24, 2022

Abstract

In this project, our group designed a low-cost sensor to collect road data, aiming at conveniently reporting the quality of seamless roads. To achieve this target, we designed an inexpensive device with an accelerometer and a GPS chip. Our device could be fixed on the bicycle with a device holder and then transmit data to the server. Machine learning models have been created to process acceleration data and judge the road quality. When abnormal patterns appeared, message where the patterns locate would be produced and sent to our display system. Locations with problems could be visualized on a city map constantly and could be accessible for users via web browsers as clients.

Contents

1 Introduction

1.1 Background

Today's transportation system has been well developed. People can find the route to a specific destination through navigation, which provides a lot of convenience. The problem is that the road condition is changing all the time, but we can't get the information of real-time road condition, such as whether there are many potholes and whether the road is flat. Because the traditional road monitoring is carried out regularly through special vehicles, it needs a lot of cost. In the blank period of monitoring, if there are serious changes in the road surface somewhere, it may affect the comfort and safety of passing vehicles, and even force vehicles to change the route, which wastes a lot of time. Therefore, we need a lower cost and more efficient seamless pavement monitoring device.

1.2 Objective

We propose a device composed of acceleration sensor, GPS, and independent power supply, with supporting mobile applications on smart phones. The device can be placed on a bicycle or vehicle and transmit the position information and corresponding sensor data to the server in real time. By filtering, processing, and extracting features of sensor data, the application can display intuitive road condition data. As a portable and low-cost monitoring device, we can equip it with many vehicles or public bicycles. In this way, the road condition of the road section can be directly obtained by the vehicles passing through the road section and provided to other users, so that everyone can obtain the real-time road condition and make a reasonable route choice.

1.3 Design Overview

Figure 1 illustrates how our system works. Our device is installed on vehicles or public bicycles and transmits the data received by the device to the server. After data processing, the results of road conditions will be marked on the map and displayed on the computer or mobile phone.

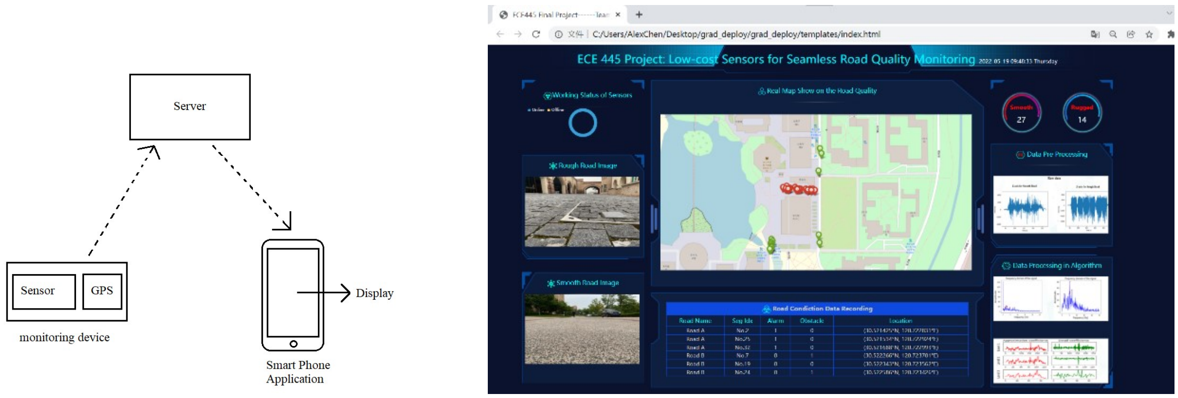


Figure 1: System Design Overview

1.4 High-level Requirements

Exclude influence of noise The action of riding should influence the results a lot. For example, riding with different speed, doing acceleration, deceleration or turning, etc. We will exclude the influence of noise by these actions on the roads.

Accuracy The result should show the real situation of the roads, including smooth roads, rough roads and obstacles. The accuracy rate of detecting rough roads and obstacles should be above 90

Speed of processing The time spent on one couple of signals should be below 100ms. We will record the length of couples of signals sent into server and record the time spent on processing. By dividing the time with length, we will get the time spent on one couple of signals.

1.5 Block Diagram

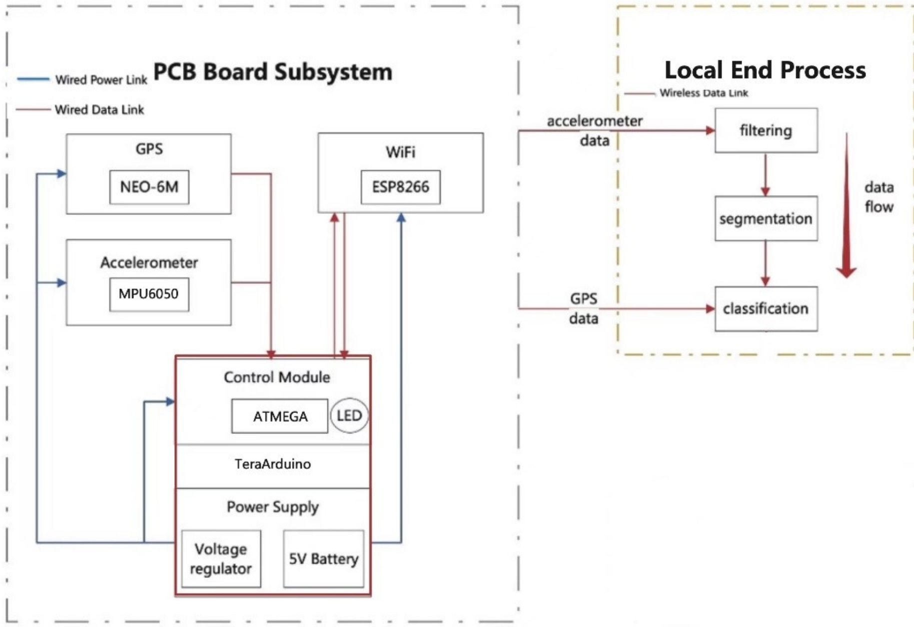


Figure 2: Block Diagram

2 Data Transmission and Result Display System

In this project, our group use python flask as the whole data transmission frame to get and process data. We use the local end to save the experiment data and use html+javascript+python to make a web page as a User Interface (UI) to display the road quality map. The following requirements are accomplished by our project design.

- The local end could automatically store and process data. Data collected from the GPS sensor module and the acceleration sensor module devices should be transmitted and stored in the local database for further processing. In this project we use txt files (log.txt) to save the raw data, the npy files (result.npy) as the intermediate process data and the json files (result.json) to save the final labeling results.
- The webpage displays a street map for some particular location range, while in this project we concentrate on the campus map.
- A red mark to indicate a road segmentation with great bound level and a green mark to indicate a road segmentation with obstacles or Speed bumps should be added to the map to indicate road quality on the campus.
- The road quality statistics measured by the user's device can be displayed on the webpage in real time.

Guided by the above design requirements, the software design work can be divided into two parts: backend and frontend.

2.1 Backend Design

The backend part consists of the database management, local end data transmission project and the data processing project. The data processing implementation will be discussed in detail in section 4. In this section, we will focus more on the data transmission.

2.1.1 Database

We use txt files to save the raw log acceleration and GPS data for the python to process in the local end disk.

2.1.2 Data fetching method

We use javascript to write our local end service. We use local API to store the data processed result by python and get these results by javasript to save as .json form in javasript file (data.js and sensor.js). And we call the .js files in the front end html page by taking them in to tight table form and map form to show the result in the html page.

2.2 Data transmission Interfaces

The interface between backend components and local end is accomplished using the Arduino programming IDE, Visual Studio Code and Jupyter Notebook. We use Arduino IDE to save the raw data and Jupyter Notebook to do the data pre-processing and processing work, among which we write python flask project to automatically deal with the produced files in the local end disk.

2.3 Frontend

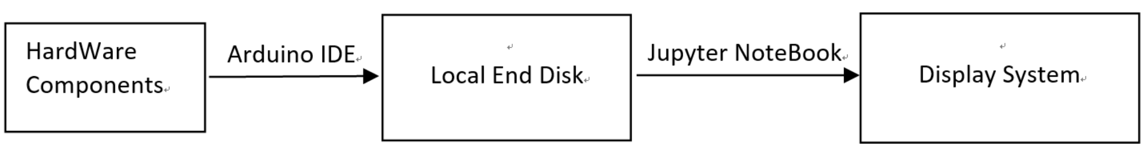


Figure 3: System Design Overview

3 Data Processing

3.1 General Processing

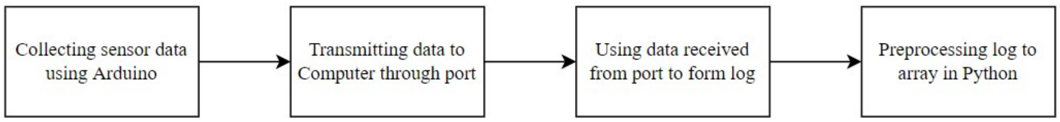
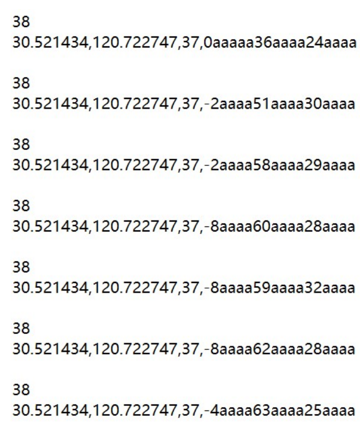


Figure 4: Illustration for data flow

The first part is Hardware part. The Arduino will integrate the signal collected from accelerometer and GPS. It will write the data to the serial. The second part is the connection between Hardware and server part. The serial data will be transmitted to computer using the wire connected. The computer can receive the data through port. The third part is the python script on computer which uses the data received from port to form log. The fourth part is the data preprocessing implemented on server. It will get the signal of accelerometer and GPS and form the array that can be accepted by the classification subsystem.

3.2 Script of Forming Log

The script of forming log keeps running. Once port receives data, the script starts get the data from port and write it on the log. Then, once the port is disconnected, the recording will finish. The format of log is illustrated below.



```
38
30.521434,120.722747,37,0aaaaa36aaaa24aaaa

38
30.521434,120.722747,37,-2aaaa51aaaa30aaaa

38
30.521434,120.722747,37,-2aaaa58aaaa29aaaa

38
30.521434,120.722747,37,-8aaaa60aaaa28aaaa

38
30.521434,120.722747,37,-8aaaa59aaaa32aaaa

38
30.521434,120.722747,37,-8aaaa62aaaa28aaaa

38
30.521434,120.722747,37,-4aaaa63aaaa25aaaa
```

Figure 5: Illustration for log

Every couple of data contains two rows. The first row is records the second. Is used to show the sample rate. The second row records the data from accelerometer and GPS. The format of this row is (longitude, latitude, second. x-acc y-acc z-acc). Though there is also time recorded, it is not updated in enough frequency, so it is not used in our system.

3.3 Preprocessing

The preprocessing part on server is a script. Once a new log is detected, the preprocessing module will start to work. It first read the log and exclude the row without useful information. Then it will get couple of rows need to be further processed. The first row, which shows the time will recorded in a list. The second rows will be processed through regular expression to get the signal from accelerometer and GPS.

After we get the raw data, we need to do resampling, because of the need of Fourier Transform and Wavelet Transform. The signal from accelerometer and GPS recorded in one second will be combined into a list.

For signal from accelerometer, we will use interpolation algorithm to resample each list to 50Hz. 50Hz is chosen because the sample rate of our raw data is 40Hz 50Hz.

Finally, the lists recorded in Ten consecutive seconds will be combined into a segment. The signal from accelerometer will form an array with dimension of (3, 500). 3 represents three axes of accelerometer. 500 represents ten seconds of signal with sample rate with 50Hz. The signal form GPS will be a tuple of longitude, latitude, which is the tuple with middle index from list combined.

The data from accelerometer will be sent to classification subsystem and data from GPS will be sent to display subsystem.

4 Classification Subsystem Design

4.1 General Processing

The classification subsystem mainly includes two parts, feature extraction and machine learning model.

A method of feature extraction should be chosen to make the abnormal pattern can be more easily recognized by model.

The machine learning model will need to use the feature extracted to classify the road as accurately as possible. In addition, the speed of processing needs to be within 0.1s for each segment.

There two tasks for the machine learning model. First, it should classify the condition of roughness of the road. Second, it should distinguish whether there is obstacle on the road. We will test the method from [1], which includes two method of feature extraction, and three machine learning model. After testing, we will choose the best combination and implement it on our server.

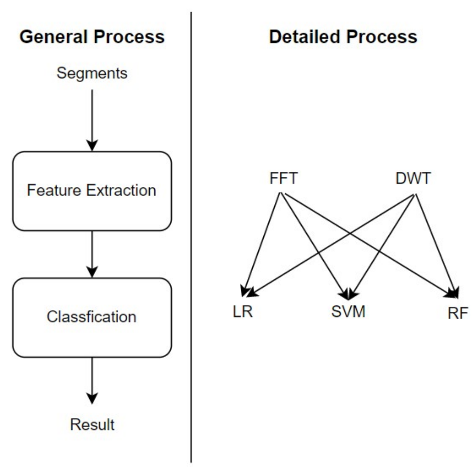


Figure 6: General process and the combinations we will test

After we get the result, the label will be labeled on the map according to the signal form GPS.

4.2 Method Description

4.2.1 Methods of feature extraction

Fast Fourier Transform

Fast Fourier Transform (FFT) is an efficient algorithm for calculating the Discrete Fourier Transform (DFT) and is the de facto standard to calculate a Fourier Transform. It is present in almost any scientific computing libraries and packages, in every programming language. [2] It can transform the signal from time domain to frequency domain. It can help us to find the main components of components with different frequency.

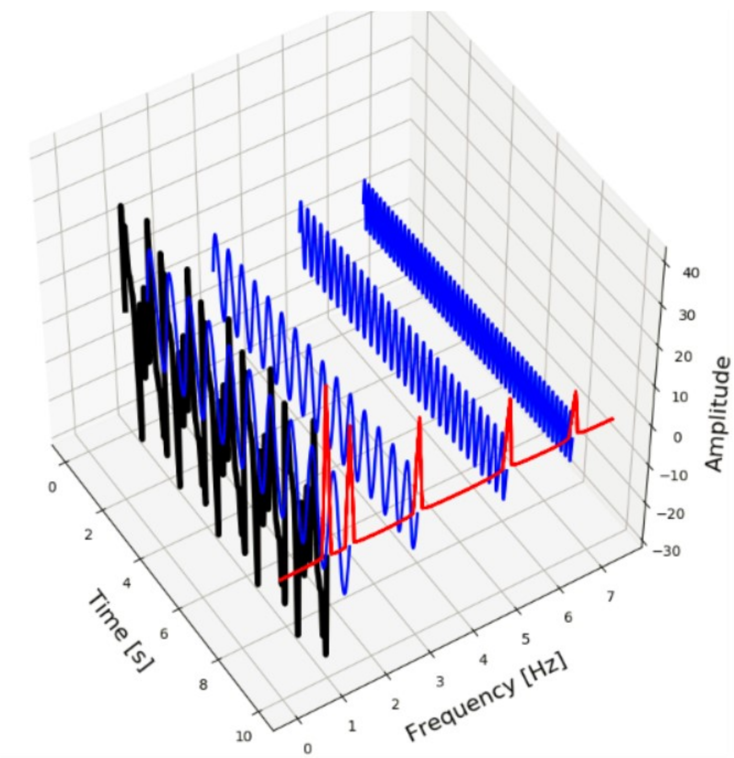


Figure 7: A signal (black) consisting of multiple component signals (blue) with different frequencies (red) form

The road quality problems can be assumed as the noise signal to accelerate signal. The road with different situation of roughness can be presented in the signal from sensors as noise with different frequency. By applying FFT, we can see the signal components with different frequencies through the wave peaks.

The result from this will be the coordinates of peaks. In addition to FFT, power spectral density (PSD), and autocorrelation functions can also be applied to transform signal to frequency domain. The coordinates of peaks vary in different methods, so these are three methods will be applied together to recognize the differences in detail.

Discrete Wavelet Transform

Though FFT can analysis the features in frequency domain, it discords all the features in time domain. In this situation, the road quality problems last in a small time can be hard to be recognized with FFT.

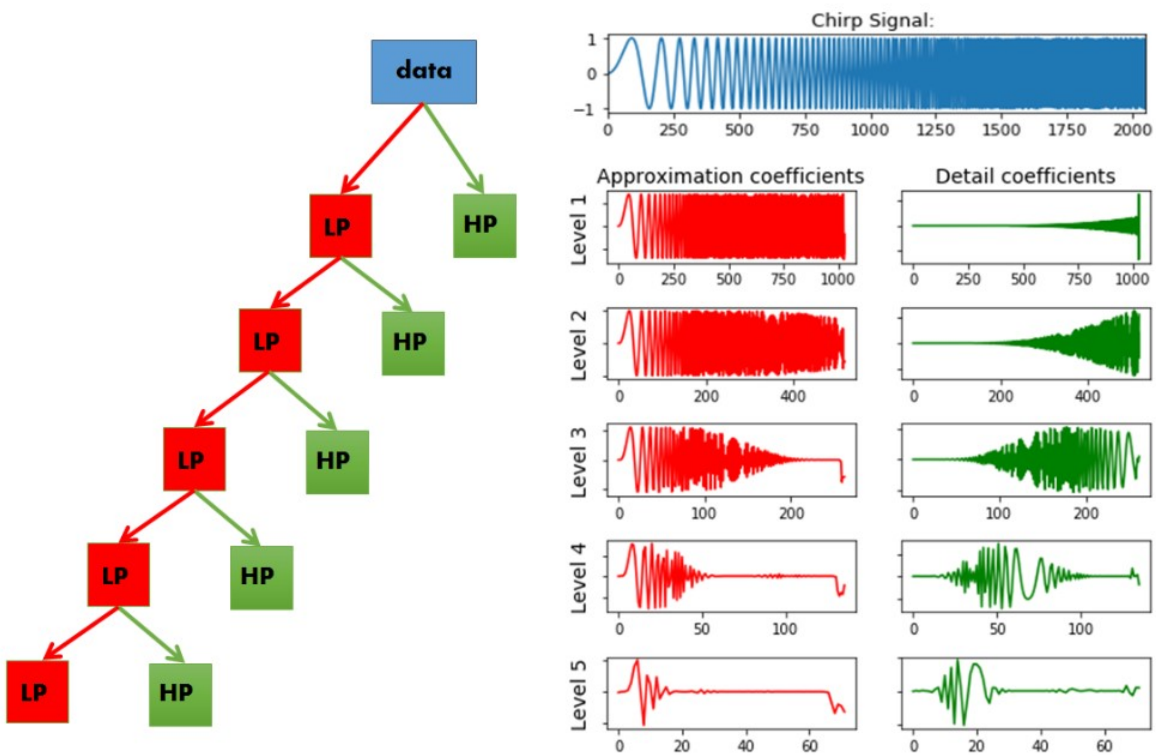


Figure 8

From [3], the figure shows the approximation and detail coefficients of the sym5 wavelet (level 1 to 5) applied on a chirp signal, from level 1 to 5. On the left we can see a schematic representation of the high pass and low pass filters applied on the signal at each level.

Discrete Wavelet Transform (DWT) solve this problem by convoluting the original signal with wavelets of different scales for decomposing the signal into several frequency sub bands. The impact on signal from problems last in a small time can be presented obviously in some sub bands.

According to [1], the result from this will be the features of some sub bands, such as entropy, 5th-percentile value, 25th-percentile value, median, mean, standard deviation, variance, number of zero crossings, etc.

4.2.2 Methods of classification

Logistic regression
The Logistic regression (LR) is a linear model for classification. It uses a logistic function to map continuous input variables X to binary output values Y.

SVM
SVM completing the classification by find a hyperplane that has the maximum margin in an n-dimensional space that distinctly divides the data points

Random forest
Random forest (RF) completing the classification by aggregating individual predictions based on various decision trees to decide the final classification.

The classification method are all basic methods, which can be completed through the library function provided by python, so I do not describe them much. The experiments on traditional methods will be the combination of two feature extraction methods and three classification methods, which is six combinations.

4.3 Implementation

For the classification, we evaluate the performance of six combinations on two tasks and we choose two best combinations. (Detailed evaluation is in verification part) For task of classifying roughness, we choose Discrete Wavelet Transform with SVM. For task of classifying obstacle, we choose Discrete Wavelet Transform with Random Forest. Once classification subsystem receive array from preprocessing, it will infer the label using two models. Then the two labels and the location from GPS will be saved as a json file. A mapping algorithm based on folium will use the information from json file to do marking on the map. The red points are the rough road and the green points are the obstacle.

Figure 9: Example for mapping

Red points are the rough road. Green points are the obstacle.

5 Hardware Design

5.1 Hardware Circuit

Our hardware part is a small device used to collect acceleration and GPS data and upload it to the server. The main components of the hardware circuit include Arduino uno R3, acceleration sensor MPU-6050, GPS module NEO-6M and 9V Ni-MH battery.

5.2 Sensor Module Description

5.2.1 Acceleration sensor

MPU-6050: The acceleration sensor module collects acceleration signal data in 3 directions to provide data for evaluating road quality. The acceleration sensor is connected with the power supply module to obtain power. And it is connected with the control module to transmit the collected data. The model of our acceleration sensor is MPU-6050, which not only includes 3-axis acceleration sensor, but also integrates 3-axis gyroscope and can communicates with MCU through IIC interface.

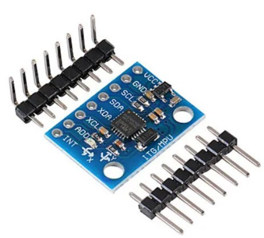


Figure 10: MPU-6050 Module

5.2.2 GPS sensor

NEO-6M: By using a NEO-6M GPS module, we can use the TinyGPS library provided by this module to get varieties of location data we need, such as latitude, longitude and altitude along with time. We can save these data and upload them into the server for further use.

Figure 11: NEO-6M Module

5.3 Microcontroller Description

We use Arduino UNO R3 as MCU, which is responsible for receiving the data of acceleration sensor and GPS and sending the data to the server. Moreover, we used STEMTera Breadboard. As an intelligent electronic bread board, it is internally compatible with Arduino UNO R3. Therefore, we can directly complete the connection between each module and MCU on this bread board.

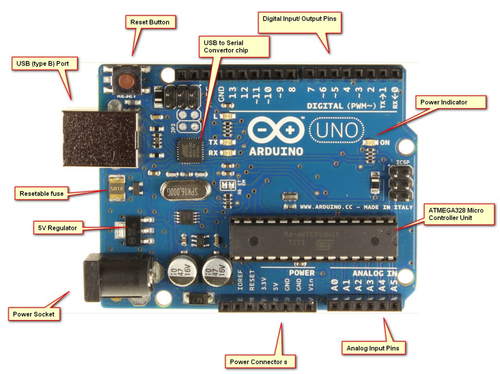


Figure 12: Arduino UNO R3

STEMTera Breadboard itself is just a physical hardware. We need to encode it so that it can correctly collect and upload data. There are many ways to send programs to Arduino UNO R3. Considering code readability, development difficulty and storage space utilization, one of the most popular methods we use is to make Arduino Integrated Development Environment (IDE). The Arduino IDE contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the STEMTera Breadboard hardware to upload programs and communicate with them.

Figure 13: STEMTera Breadboard

5.4 Power Supply

Our power supply is a 9V nickel-metal hybride battery with 360mAh capacity. This capacity can ensure a long working time of the device for a vehicle's test period on a road. And because the STEMTera Breadboard itself has 3.3V and 5V interface, we do not need a voltage regulator.

Figure 14: 9V Ni-MH battery

5.5 Module Interfacing Details

Arduino UNO R3 connects with other modules to collect and upload data. This section shows all interface details between Arduino UNO R3 and each module. The following figure shows the pin diagram of the microcontroller.

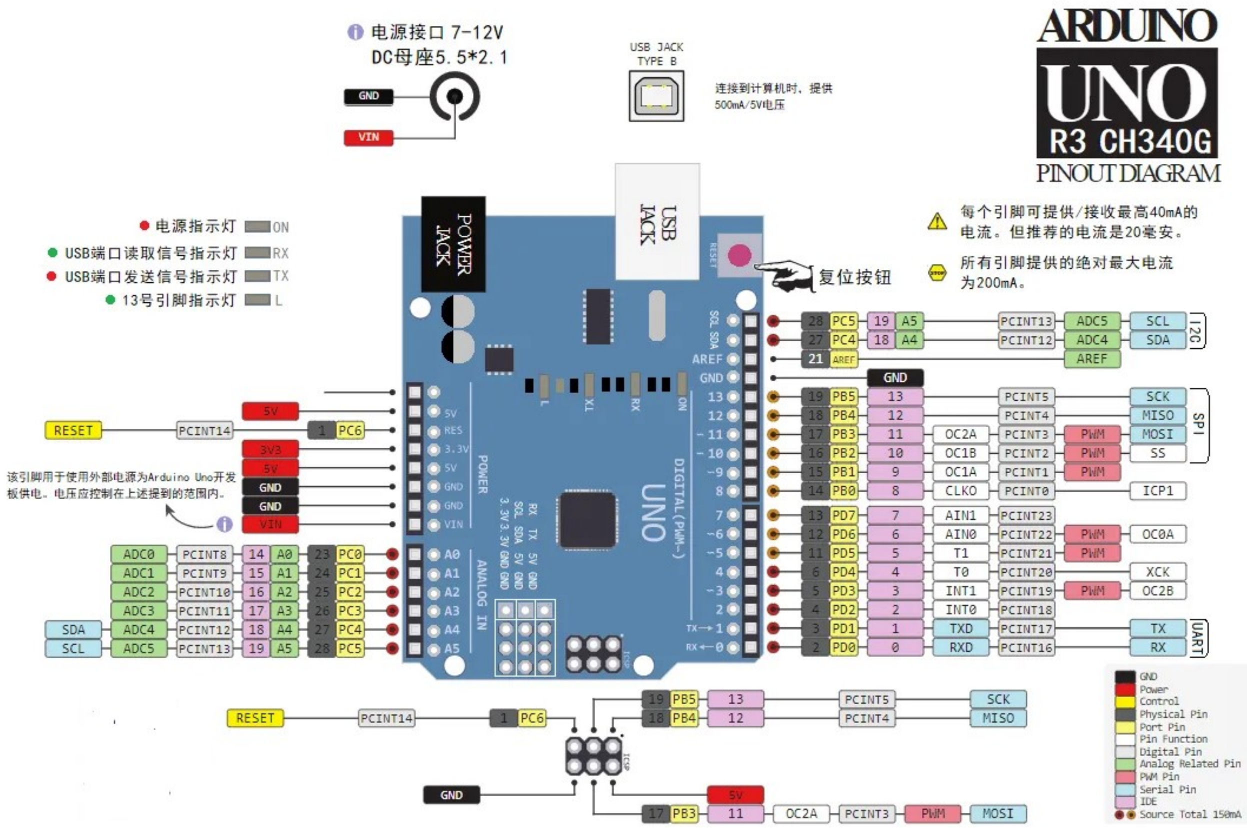


Figure 15: Pin Diagram of Arduino UNO R3

NEO-6M Module

NEO-6M GPS module has four interfaces to be connected. VCC interface is connected to 5V, GND interface is connected to GND, and TX and RX interfaces for receiving and transmitting data are connected to 4 and 3 respectively on Arduino.

MPU-6050 Module

MPU-6050 module has four interfaces to be connected. VCC interface is connected to 5V, GND interface is connected to GND, and SCL and SDA interfaces for data transmission and clock are connected to A5 and A4 respectively on Arduino.

This figure is detailed device interface schematic.

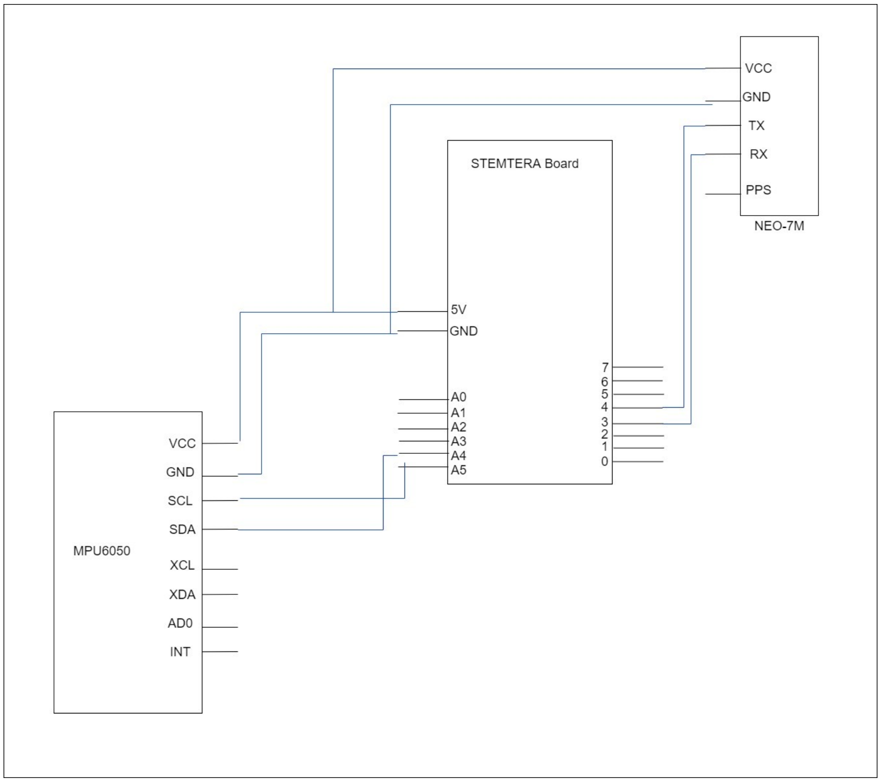


Figure 16: Device Interface Schematic

5.6 Physical Device Connection

According to the device interface diagram, connect the module with the bread board, and fix the GPS ANTENNA on the STEMTera Breadboard. We use wire jumpers to connect and try to complete our device in the most concise way. The following is the picture of our device.

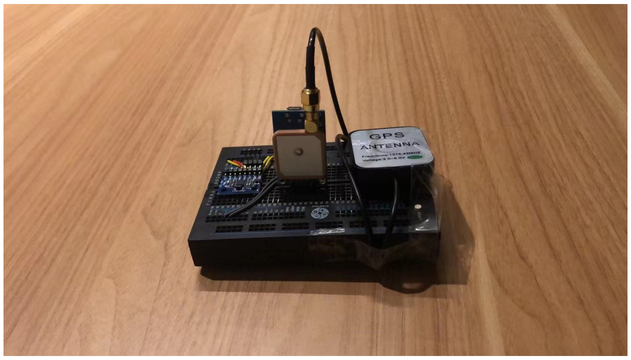


Figure 17: Picture of the Completed Device

6 Mechanical Design

Our device needs to be fixed on the bicycle to collect data. Therefore, in addition to its own hardware, we also need a device holder to fix the device.

This is to ensure that our hardware will not fall off in rough road conditions and pass-through rough road, for example the speed bumps. In consideration of 3D printing and commercial products, we finally chose commercial products because of its better effect and mature design. This is a holder for fixing the device through bolts and silicone strips. We can fix the device firmly on the device holder by adjusting the length of the intermediate bolt. After installing and testing, we found that it has a good fixing effect. The following diagram is the picture of the device holder.

Figure 18: Picture of The Device Holder

7 Verification

7.1 Feature Extraction Analysis

In this part, we will verify the effectiveness of the method of feature extraction.

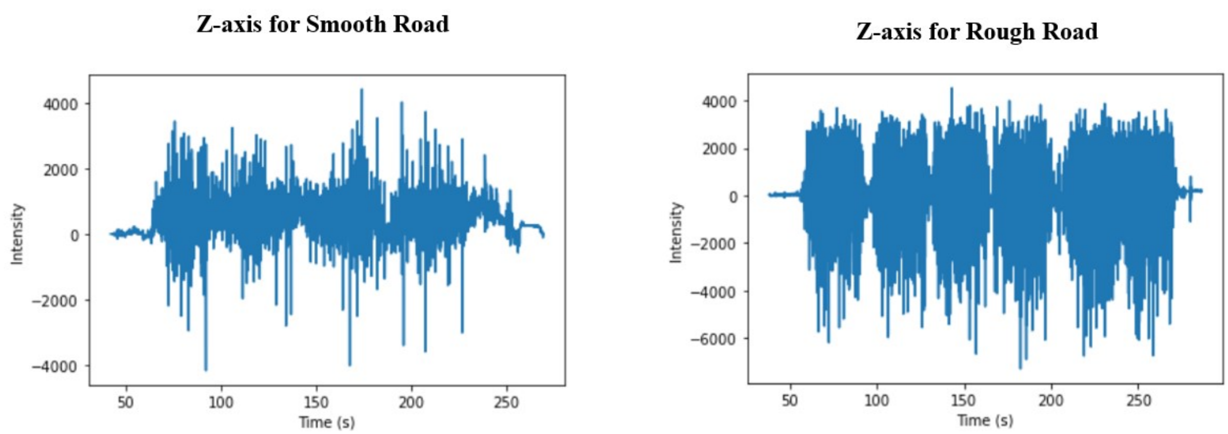


Figure 19: Comparison of raw data

From the raw data for the Z-axis, we can the see that though the magnitude of the maximum intensity is almost the same, the frequency of such high intensity is different. It is natural for us to extract the feature in frequency domain, so the first method of feature extraction is Fourier Transform.

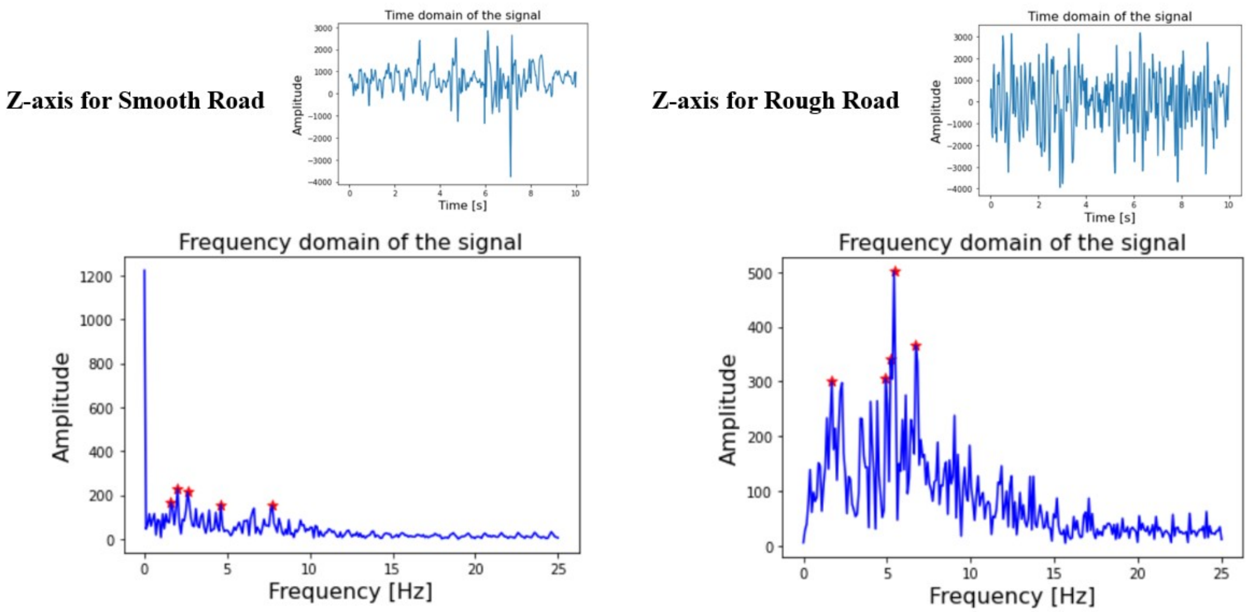


Figure 20: Comparison of data after Fourier Transform

By comparing the figure after Fourier Transform, we can see the pattern is obviously different. We can use the peaks in frequency domain as the features since roughness can be assumed as the noise signal to accelerate signal. The road with different situation of roughness can be presented in the signal from sensors as noise with different frequency.

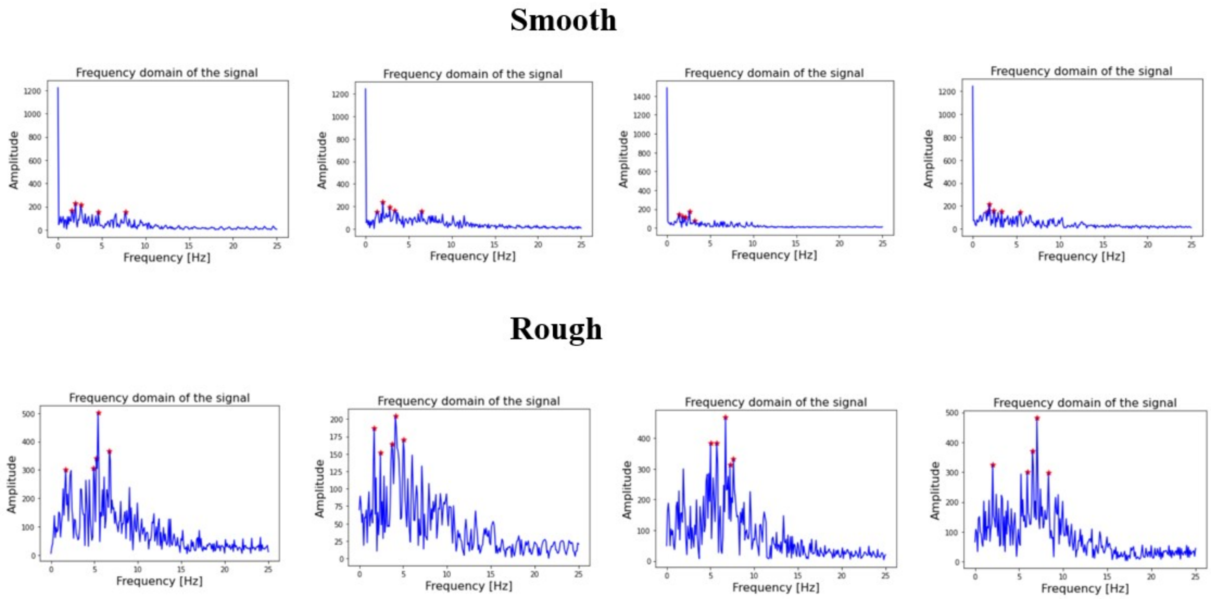


Figure 21: Comparison of data with different roughness

Comparing several figures after Fourier Transform. We can see that the Fourier Transform can help us easily distinguish the roughness, because the difference in pattern occurs in many couples of data.

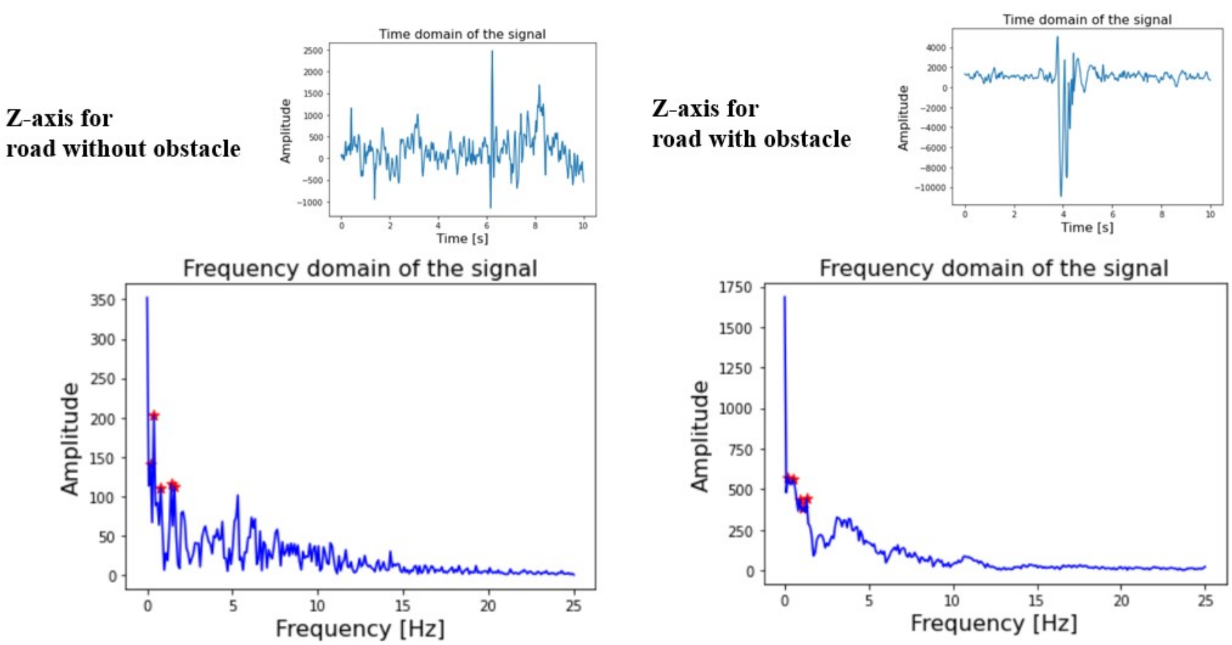


Figure 22: Comparison of data after Fourier Transform for task of distinguishing obstacle

However, when we try to use Fourier Transform to distinguish obstacle, we find that there is no obvious difference in pattern. The pattern from raw data is a mutation. In the frequency domain, it will be a peak in the low frequency. However, for the road without obstacle, it can also have mutation because of the acceleration, deceleration, etc. In this situation, it can also has peaks in low frequency.

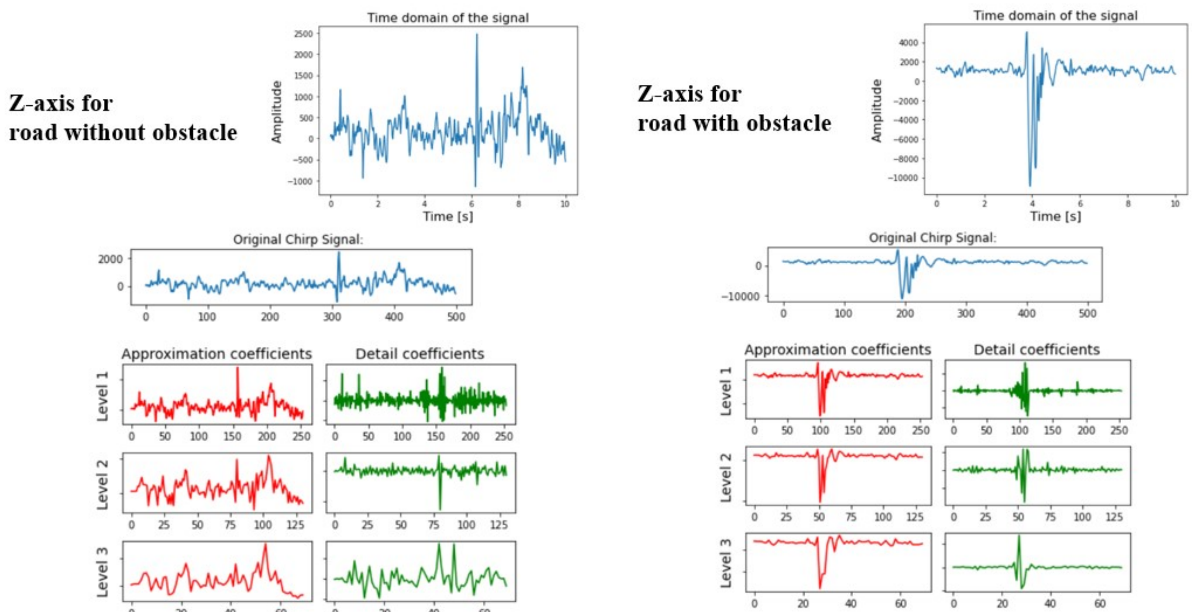


Figure 23: Comparison of data after Wavelet Transform for task of distinguishing obstacle

Wavelet Transform can help us better extract the feature. Different from Fourier Transform, which discards the time sequence, Wavelet Transform keeps the time sequence by convolutes with a non-periodic signal. It keeps resolution in frequency domain by convoluting with different frequencies. We can see the mutation from raw data is kept well in the figure after Wavelet Transform. We will use entropy, 5th-percentile value, 25th-percentile value, 75th-percentile value, 95th-percentile value, median, mean, standard deviation, variance, RMS, number of zero crossings, number of mean crossings of four sub bands as the feature.

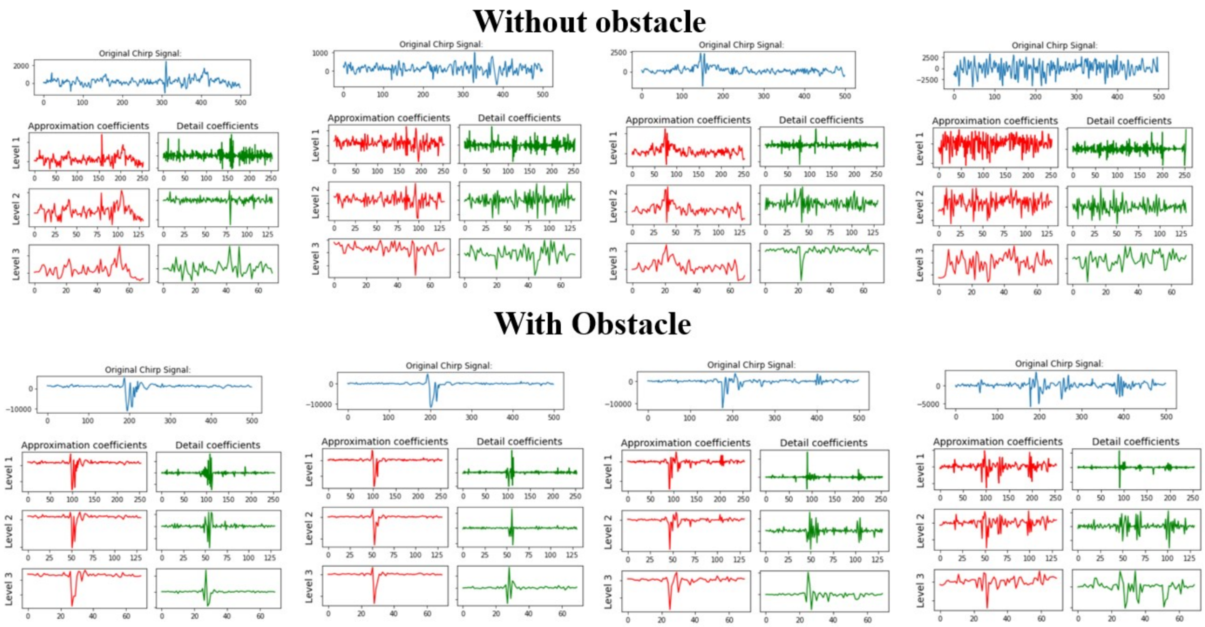


Figure 24: Comparison of data with and without obstacle

By comparing couples of data, we can see the difference in pattern is obvious in most of data. However, if the mutation is not big in magnitude, the pattern will become less obvious, which is likely to be classified as no obstacle.

7.2 Data Set Analysis

Since our data is hard to collect, we do some data enforcement to make our data set more robust. We do not easily split the raw data by 10 seconds but moving the clipping window along the time-series with the step of 1 second. In this situation, we can get ten times of data than the original splitting method. For signals with obstacles, the data is hard to splitting, because the window need to contain the mutation and we do not know the duration of the mutation. Our samples for signals with obstacles is much fewer than the ones without obstacles.

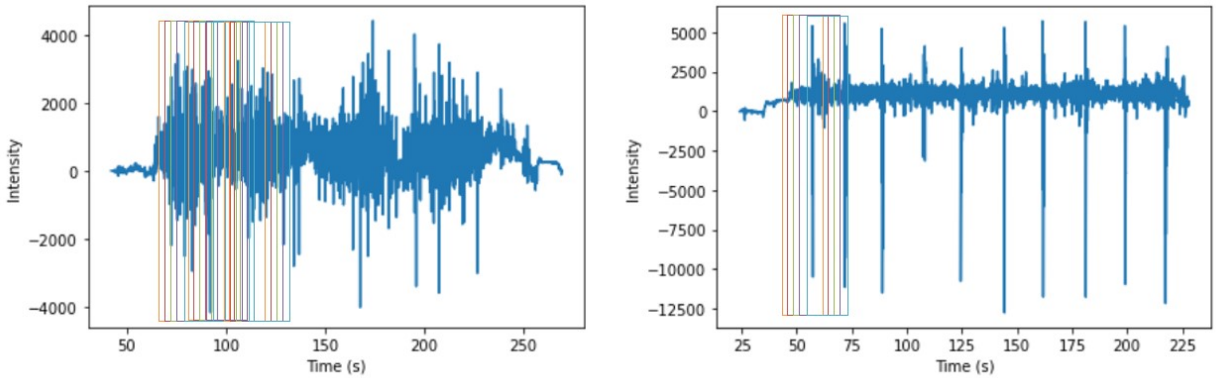


Figure 25: New method of splitting data

7.3 Model Analysis

7.3.1 Classification for roughness

To improve the robustness, the testing set and training set is different. The testing set contains the road that the training set has never seen, which evaluate the performance of generalization of our model.

Method of Feature Extraction	Classification Model	Accuracy	Precision (TP/(TP + FP))	Recall (TP/(TP + FN))
Fast Fourier Transform	Random Forest	88.4%	100%	66.6%
Fast Fourier Transform	Logistic Regression	95.8%	82%	100%
Fast Fourier Transform	SVM	100%	100%	100%
Discrete Wavelet Transform	Random Forest	83.3%	100%	58.1%
Discrete Wavelet Transform	Logistic Regression	95.4%	82%	100%
Discrete Wavelet Transform	SVM	100%	100%	100%

Table 1: The result for evaluating performance of classification for roughness

7.3.2 Classification for obstacle

To improve the robustness, the testing set and training set is different. The testing set contains the obstacle that the training set has never seen, which evaluate the performance of generalization of our model.

Method of Feature Extraction	Classification Model	Accuracy	Precision (TP/(TP + FP))	Recall (TP/(TP + FN))
Fast Fourier Transform	Random Forest	96.4%	67.6%	100%
Fast Fourier Transform	Logistic Regression	82.1%	59.5%	32.4%
Fast Fourier Transform	SVM	87.7%	72%	45.8%
Discrete Wavelet Transform	Random Forest	99.4%	94.5%	100%
Discrete Wavelet Transform	Logistic Regression	96.5%	81.1%	85.7%
Discrete Wavelet Transform	SVM	96.7%	89.2%	82.5%

Table 2: The result for evaluating performance of classification for obstacle

7.4 Speed Analysis

We test with a log with duration of 225 seconds. The time spent on processing it is within 2 seconds. For 225 seconds, there are 22 segments, so it can be proved that the time spent on processing one segment is within 0.1 second, which reaches our expectation.

7.5 Block Diagram

We test the tolerance on noise cause by special behavior of riding such acceleration, deceleration, and turning. The path of riding is show below.

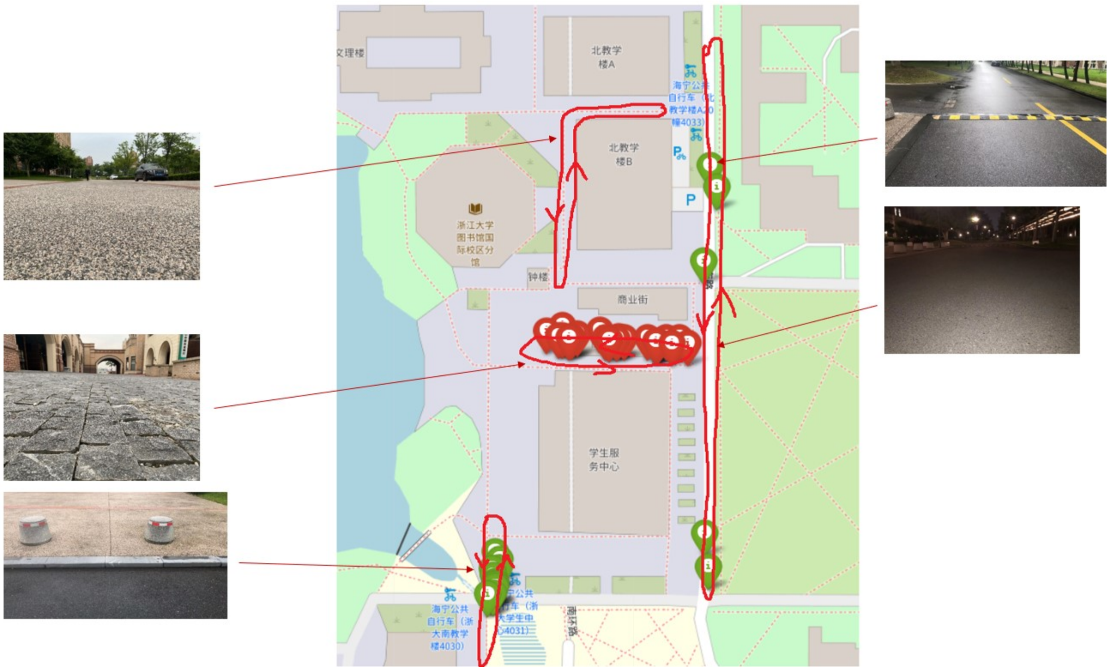


Figure 26: Path of riding for test

Our path includes the acceleration, deceleration, and turning. For each path, the smooth road is not labeled. The rough road is labeled as red. The road with step is labeled as green. The road with speed band is labeled green. There is an error on a road without obstacle. We can see that the noise do not influence the accuracy of our classification model in a great degree.

8 Cost

8.1 Hardware Component Cost Analysis

Name	Description	Cost per unit	Number	Others
MPU6050	Sensors Module	￥8.4	1	N/A
ESP8266	Wi-Fi Module	￥28	1	N/A
NEO-6M	GPS Module	￥24	1	N/A
Stm32F103C8T6	Control Module	￥36.88	1	N/A
18650 Battery	5V	￥14.55	2	N/A
GPS SMA	GPS signal enhancement	￥12	1	N/A
BreadBoard Wires	Circuit Wires	￥4/140	140	N/A

Table 3: Hardware Component Cost

8.2 Mechanical Component Cost Analysis

Device	Description	Cost per unit	Rent Hour	Total Cost
Rent Bicycle	Tested Vehicle	￥2/hour	30	￥60
Five-claw handlebar	Fixing device	￥28	1	￥28

Table 4: Mechanical Component Cost

8.3 Labor Component Cost Analysis

Name	Hourly Wage	Total Hour	Total Cost	Other
Yuhang Chen	￥90	120	￥10800	N/A
Yichen Li	￥90	120	￥10800	N/A
Jingyao Dai	￥90	120	￥10800	N/A
Yihang Yang	￥90	120	￥10800	N/A

Table 5: Labor Component Cost

8.4 Total Cost

Our system should consists of the above components installed on the public bike. But due to the time limit and challenge, only several samples are used for the functional demonstration. The cost for n unit(s) is 43200 + 215.83n, where n is the number of our system devices.

9 Conclusion

9.1 Accomplishments

In this project, we have successfully built a Low-cost Sensor for Seamless Road Quality Monitoring. For the hardware design, we have a highly developed design of our mobile road quality monitoring device, which includes the design of the hardware circuit of connecting a TeraArduino Board, a GPS sensor module and an acceleration module, the Arduino Board microcontroller programming, the sensors driving programming as well as the WIFI module driving programming. For the software design, we have tried to do the data transmission by the WIFI module and we succeed in Esp8266 STA mode to send data through personal hotspot to the local end without USB wires, while due to esp8266 transmission limits we abandon it at last. And we use python flask to write the whole project frame to automatically process data received by our local end. The processed results could automatically making marks on a map and a html page is automatically made by the processing data. For the data processing algorithm, we implemented Fourier-transform algorithm and formulated the inference problem as a matrix completion problem. And we extract varieties of useful features to do the data training and labeling. Our project could label the poor road quality with high degree of bound level as the red mark on a map. Also, we could label the poor road quality with obstacles or Speed bump as the green mark on a map. On the mechanical design part, we simply use a Five-claw handlebar to fix our device system on the front end of a bicycle.

9.2 Uncertainties

Our data set is built based on the campus road, which is not strong enough to ensure the certainty of applying on the other roads that are out of campus. Due to the data set limit, there are uncertainties includes the type of rough roads, the varieties of bound levels, etc. might appear in the future tests.

9.3 Ethical considerations

Our design needs to collect the data of location, time, and different directions of acceleration during the procedure of riding a bicycle. In the development and use of our project, we must avoid violating ethical breaches. Since our device has GPS positioning function and position is a kind of personal privacy, we must comply with 7.8 IEEE Code of Ethics Section1.1 [4] and ACM Code of Ethics 1.6 [5] to strictly protect the privacy of users, collect only the minimum amount of necessary personal information, prevent accidental data leakage, and timely disclose factors that may endanger the public or the environment.

Moreover, because our data will help users determine the driving route, we must comply with ACM code of ethics 2.9 to establish a safe system to avoid problems caused by system vulnerabilities or unexpected data tampering. The wrong route may affect driving comfort and even safety. We should conduct due diligence and provide appropriate guidance and remedial measures in case of problems.

9.4 Future work

For hardware design, currently we do not use esp8266 WIFI module due to the wireless challenge. The WiFi transmission by esp8266 lower the transmission frequency. The Sample rate of the data after WiFi transmission decreases with a great degree (40 50Hz to 10 20Hz). And ESP8266 need to first tell the server to receive data then send the data, which decreases the transmission frequency. We also learn the 4G module could be a better choice for transmitting data during the moving process and could provide more stable links, which worth trying in future.

For user interface, a mobile app can be developed for real use instead of the current webpage, which just show the processed results.

For processing algorithm, we could do the following two steps to improve the ability to process data:

Cancel the segmentation with same distance
With applying Fourier Transform and Wavelet Transform, the duration should keep the same. Otherwise, the difference in parameters of Fourier Transform and Wavelet Transform may reduce the accuracy of models.

Increase the resampling part
Get the sensor data in one second according to log. Use interpolation algorithm to resample it to 50Hz. It confirms the accuracy of Fourier Transform and Wavelet Transform.

For data set building, we could improve in the data set labeling. We could use a camera to record the road quality. Comparing to labelling by recording the timestamps, the camera can tell the road quality more accurately and conveniently. The quality of data decides the accuracy of model in great degree.

References

[1] C. W. et al, “An Automated Machine-Learning Approach for Road Pothole Detection Using Smartphone Sensor Data,” *Sensors*, vol. vol.20,no.19, no. 5564, 2020.

[2] A. Taspinar. “Machine learning with signal processing techniques.” (2018), [Online]. Available: http://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques (visited on 05/24/2022).

[3] ——, “A guide for using the wavelet transform in machine learning.” (2018), [Online]. Available: http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning (visited on 05/24/2022).

[4] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 05/24/2022).

[5] ACM. “”ACM Code of Ethics and Professional Conduct”.” (2016), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 05/24/2022).