

Classification of types of road damage using machine learning and a smartwatch

Martijn Renssen
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

ABSTRACT

The goal of this research is to investigate how accurately machine learning algorithms can classify different types of road damage using accelerometer data from a smartwatch. The accelerometer data is not collected with a real smartwatch, but with a wearable sensor made for motion capture. The types of road damage that were tested with are potholes, cracks and crocodile cracks in asphalt roads only.

Keywords

Road damage, Classification, Machine learning, Smartwatch, accelerometer

1. INTRODUCTION

The infrastructure is a big part of our society and a big part of the every-day life of many people. Roads are used a lot every day by people to get to work or to go home, for freight traffic, by tourists, etc. Because the roads are used so much the throughout the entirety of every day, the safety of the roads is very important. Damaged roads are a hazard for drivers because they can cause accidents, and therefore it is very important that damaged roads are repaired as soon as possible.

Detecting and reporting damaged parts of roads are one of most important tasks of road maintenance, because roads can only be repaired if it is known what type of road damage has to be repaired at which part of a certain road. In the Netherlands it is possible for a civilian to help bring attention to the existence of road damage somewhere by making a phone-call to the "Directorate-General for Public Works and Water Management" (in Dutch: Rijkswaterstaat). However having to make a phone-call to do this may be discouraging to people, and it is not a safe thing to do while driving. Therefore an alternate way of helping the discovery of road damage could be beneficial for civilians that want to help and for the general safety of the roads.

Nowadays almost everyone has a smartphone, and smartwatches are also becoming more and more popular. These mobile devices can be used to collect accelerometer- and GPS-data, and if machine learning can be used to reli-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

30th Twente Student Conference on IT Febr. 1st, 2019, Enschede, The Netherlands.

Copyright 2019, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

ably identify different types of road damage from the accelerometer data, the process of damage detection could be automated. This will speed up the rate of damage discovery and repair, which is beneficial for both the government and the drivers.

Currently research is already being done on technology that can automatically detect road damage. At the University of Waterloo, researchers have developed an artificial intelligence system capable of locating potholes using images collected with vehicle-mounted cameras[3]. Another research project, called ePave, researches the use of self-powered wireless sensors that are to be placed under the surface of the road to monitor the road quality[2]. The main difference between the proposed solution and the previously mentioned solutions is that the proposed solution does not require the installation of additional technology which may be expensive and require maintenance.

2. RESEARCH QUESTIONS

The main research question of this research is:

How accurately can machine learning algorithms classify different types of damage to asphalt roads by using accelerometer data collected from a smartwatch while driving?

To explore the accuracy of the algorithms in more detail, and to test the influence of different circumstances while driving, the following sub-questions are addressed:

1. How does the position of the smartwatch affect the accuracy of the algorithms?
2. Which machine learning classification algorithm has the best overall accuracy?

3. RELATED WORK

Research has already been done several times on the use of the accelerometer of a smartphone to detect road damage.

An example is the research that was published in 2011 by Mednis et al.[7] in which they proposed and tested four different algorithms that they used to classify large potholes, small potholes, pothole clusters, gaps and drain pits. They collected the data with the accelerometer of different smartphones, and they ran their classification algorithms on the smartphones in real time. The algorithms they developed are based on a certain threshold value. When certain thresholds are exceeded, the algorithm can make predictions based on that. With one of their algorithms, they were able to achieve an accuracy of over 90%.

In a more recent research (2016), Gawad et al. [4] researched the use of machine learning on accelerometer data from smartphones to detect road damage. In this

research a neural network was used to dynamically calculate a threshold based on the readings of the accelerometer. This threshold is then used to decide if the car is driving over an anomaly or not. In this research they do not classify the different types of damage, but it is only used to detect that the road is damaged. The accuracy they were able to achieve was at least 70%.

Carbal et al. (2018) [1] did research on the use of four different classification algorithms (Support Vector Machine, Hidden Markov Model, Residual Network and K-Nearest-Neighbors with DTW) to classify between paved and unpaved road, and in case of a paved road also the classification of damaged or undamaged road. As input for the algorithms, accelerometer data from a smartphone was used. A sliding window of two seconds was applied to the accelerometer data, and 130 features were calculated for each window. With both the Support Vector Machine and the K-Nearest-Neighbors with DTW algorithms they were able to classify damaged road from undamaged road with an accuracy of up to 100%.

Apart from the use of the accelerometer, also the use of the camera of smartphones has been researched. In 2018, Maeda et al. [6] published their research on the use of the camera of a smartphone in combination with deep neural networks to classify different types of road damage. In their research, they were able to classify 8 classes of road damage with up to 77% accuracy. An advantage of using images over vibrations to classify road damage is that by using images, also damage to paint of traffic signs on the road, such as cross walks, can be detected. A disadvantage is that shadows can cause false classifications, and if it is raining, the phone will not have a clear vision of the road.

The research published in this paper is different from previous research because the use of a smartwatch is investigated instead of a smartphone, and the goal of the machine learning algorithms is to not only detect that the road is damaged, but also to determine what type of damage is present based on accelerometer data.

4. METHODOLOGY

4.1 Data collection

For this research, a new had to be data-set generated, since a data-set that meets the requirements for this research was not available.

4.1.1 Used equipment

Because of the limited amount of time in which the research had to be performed, and in order to make the collection of data less complex, two wearable *ProMove-mini*[5] sensor devices were used instead of a real smartwatch. To record the data, only the accelerometer of the *ProMove-mini* was used. A sampling rate of 100 Hz was chosen because this appeared to be the most common sampling rate of the accelerometers used in smartwatches. After collecting data with this device, the data was saved to a CSV-file to be able to easily import and manipulate the data later. These CSV-files contain a column for a timestamp, and a column for each axis of the accelerometer i.e. x, y, and z axis.

4.1.2 Test drives

All of the data used in this research was collected on asphalt roads while driving at a speed between 50 and 70 km/h. During the collection of the data, a voice recorder was used to be able to call out when the car drove over a damaged piece of road, or over other important noticeable things such as speed bumps or a piece of non-asphalt road.

The collection of the data was done in two times, but the type of roads and the speed were kept as similar as possible. During the first ride, one of the sensors was firmly strapped to the left wrist, since this is where most people wear their watch, and the other sensor was placed in the pocket of the driver to try to simulate an arm that is not holding the steering wheel, but is lying on the driver's lap. During the second ride, one sensor was again placed on the left wrist, and the other sensor was attached to the dashboard of the car with the idea that this would pick up the exact vibrations coming from the car while driving.

4.1.3 Labeling the data

Labeling the data was done by using the recorded voice during driving, and a simple tool developed in Python for this specific research. In the tool, two CSV-files can be opened and visualised at the same time so that the data from both sensors can be seen at the same time. After clicking on two points in the graph, a new label with the selected name is added between the chosen points. After the labeling is done, the labels can be exported to a CSV-file. Figure 1 shows what the tool looks like. The label types that were used are: 'normal road', 'crack', 'crocodile crack', 'speed bump' and 'excluded'. The label 'excluded' was used to label parts of the data where the car was not driving over a normal asphalt road.

4.2 Processing the sensor-data

The processing of the sensor data and the implementation of the machine learning were done in Python, since it offers many public frameworks which can be used for this type of research.

4.2.1 Importing the data into Python

To be able to manipulate and use the data from the sensors, the data needed to be imported into Python. To do this, the Python library *pandas* [8] was used, which can load a CSV-file into a *DataFrame* object, which can be seen as a table that can be accessed by row- and column index, but it also allows for easy selection based on values.

4.2.2 Calculating the total acceleration

After the sensor-data was loaded, the three acceleration axes were combined into one value representing the total acceleration. This was done according to formula 1. In this formula, a_x , a_y , and a_z stand for the acceleration measured on the x, y, and z axis.

$$a_{total} = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (1)$$

After this, to compensate for possible differences in the calibration of the two sensors, the data was normalized by subtracting the mean of the entire measurement from each individual data-point.

4.2.3 Rolling window and statistics

Since a single data-point does not characterise a certain vibration or movement, a rolling window was used to be able to train the machine learning algorithms on sets on data-points. A *pandas DataFrame* offers a function that can generate a rolling window using the values it contains, given a certain window size. This window shifts 1 value after each window. The window size that was chosen was 50, which corresponds to half a second of data, since the sampling rate was set to 100 Hz. For each window of 50 values, by using the built-in functions of *pandas*, the following statistics were calculated: mean, maximum, minimum, median, standard deviation, 25% quantile, 75% quantile, kurtosis, skewness, and root mean square. The value of

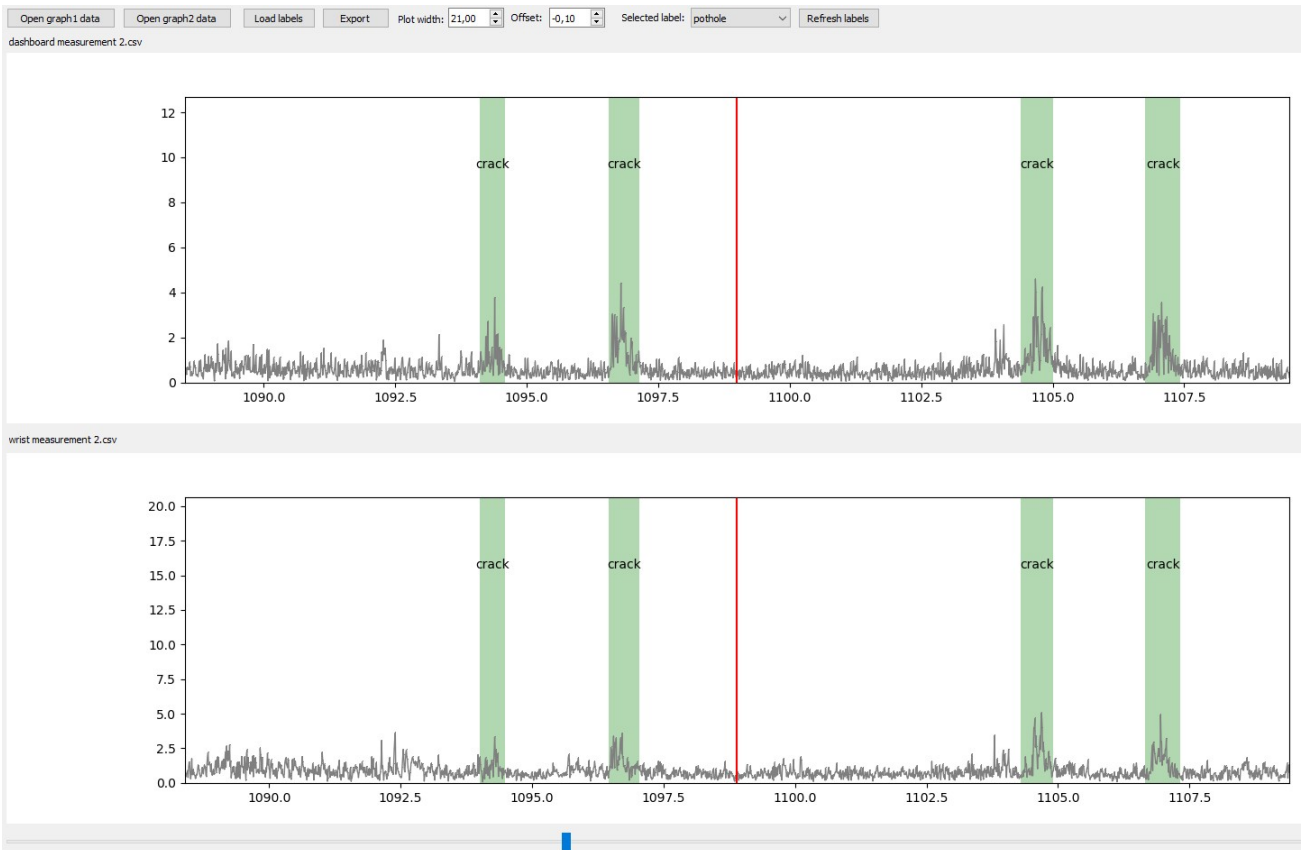


Figure 1. Screenshot of the labeling tool.

each of these statistics was added as a new column at the last data-point of each window. This means that the row of the 50th data-point contained the statistics of the first window, and that row 51 contained the statistics of the second window and so forth.

4.2.4 Adding the labels to the data

Adding the labels to the data-points was done for each label by adding the label as a new column to each data-point between the timestamp of the begin of the label plus 0.49 second (the duration of 49 data-points) and the timestamp of the end of the label. The 0.49 second is added to make sure that the label starts at the row that contains the statistics of the first window that only contains data of the type of the label. After all the labels were added, the rows that had no label, and the rows that were labeled as 'excluded' were removed. The rows without a label contain statistics of windows that contain data-points from two different label types, such as a crack and normal road.

4.3 Machine learning

The implementation of the machine learning was done using the open-source library *Scikit-learn*[9]. *Scikit-learn* offers several types of classification algorithms, of which the *Linear SVC*, *KNeighbors* and *Naive Bayes* algorithms were tested. These algorithms were chosen based on a flow-chart[10] from *Scikit-learn* in which different circumstances lead to different algorithm recommendations. The circumstances of this research lead to the mentioned classifiers.

4.3.1 Splitting the data into train- and test sets

To split each *DataFrame* into a train- and a test set, the *DataFrame* was first split into 100 parts, then each of those parts were split into 70% train set and 30% test set. After this, all the separate parts of each set were concatenated to form one train- and one test set. The first split into 100 parts was done to make sure that data from many parts of each measurement would be in both sets.

4.3.2 Training and testing the algorithms

To train each algorithm, only the label column and train set and the the columns containing the statistics needed to be passed to the algorithm. When the algorithms were trained, the statistics columns of the test set were given to the algorithm, which then made a prediction of the label that belonged to each row of the test set.

4.3.3 Evaluating the performance of the machine learning algorithms

The performance of the machine learning algorithms was determined for each different label type by calculating the *F1 score*, which ranges from 0 to 1, with 0 being the worst and 1 the best. The *F1 score* is calculated using formula 2.

$$F_1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \quad (2)$$

In this formula, *precision* is the number of correct positive results divided by the number of all positive results returned by the classifier, and *recall* is the number of correct positive results divided by the number of all samples that should have been identified as positive.

5. RESULTS

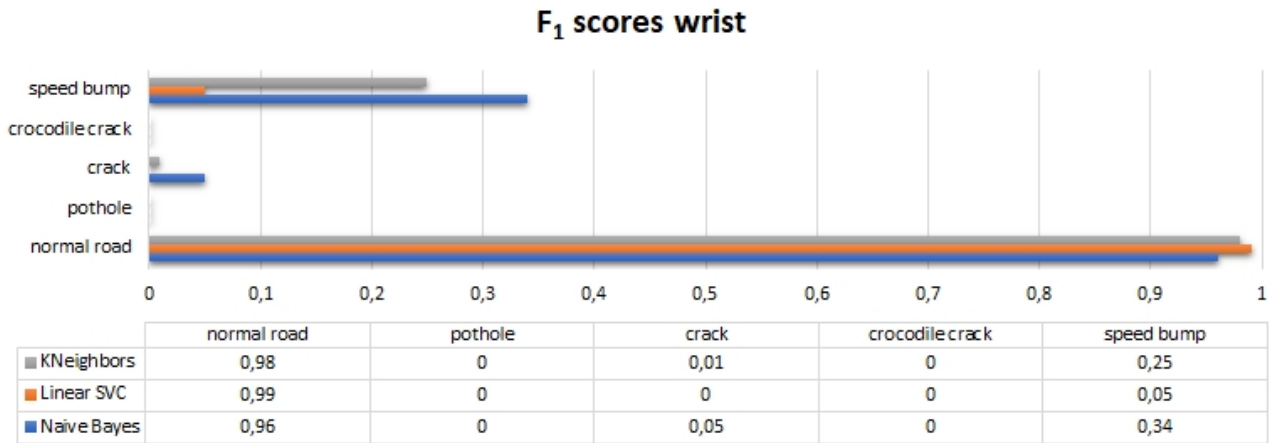


Figure 2. F₁ scores of the wrist sensor.

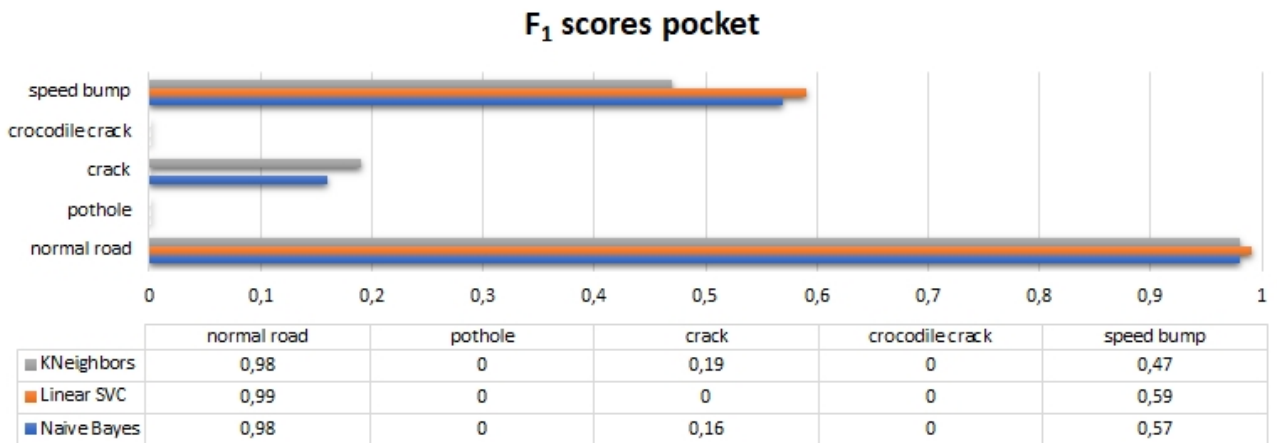


Figure 3. F₁ scores of the pocket sensor.

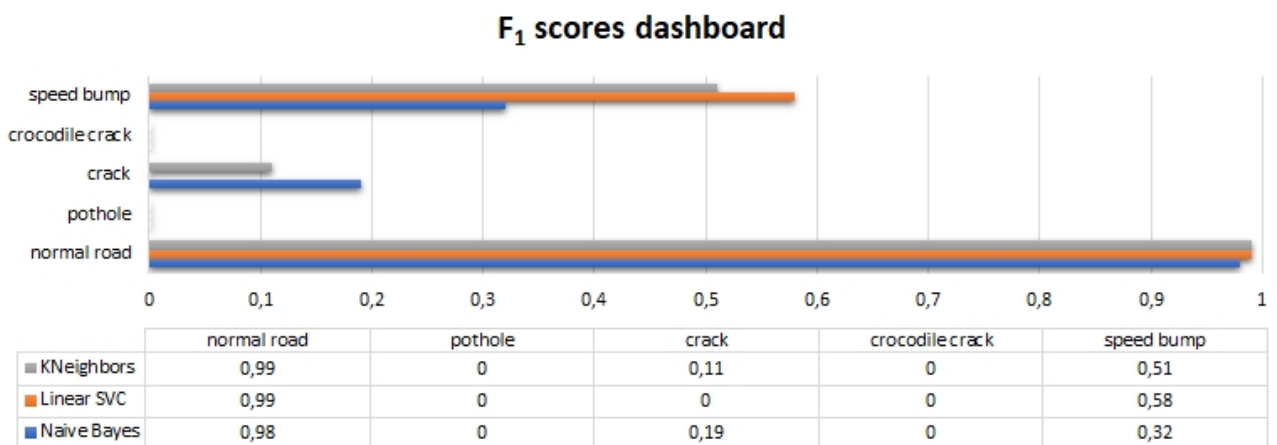


Figure 4. F₁ scores of the dashboard sensor.

After training and testing each of the classifiers on each separate data-set, the results shown in figure 2, 3, and 4 were obtained.

By taking the average of all the F_1 scores of each classifier the following overall F_1 scores were calculated:

KNeighbors: 0.299

Linear SVC: 0.279

Naive Bayes: 0.303

6. DISCUSSION

Based on the research of Carbal et al. in which they managed to achieve an accuracy of up to 100% by using accelerometer data from a smartphone and machine learning algorithms, the results of this research may seem a bit disappointing. However, their research is focused on detecting if the road is damaged or not, and not on what type of damage, or obstacle it is. Also the classification algorithms and pre-processing methods that were used are different.

None of the algorithms were able to correctly classify crocodile cracks or potholes, for any of the positions of the sensors. A possible explanation for this would be that the vibration patterns of these types of damage are very similar to the vibration patterns of the roads that were used for testing. Figure 5 and 6 show the patterns of these types of damage. The parts marked in green belong to the damaged road, and the unmarked parts are undamaged parts of the road. When looking at these graphs it seems that there is little to no difference between the patterns of the damage and the patterns of other parts of the road.

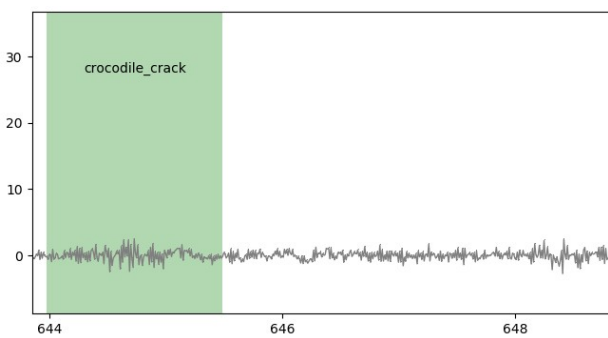


Figure 5. Vibration pattern of a crocodile crack.

This could also be the cause for the low F_1 scores for the cracks. Figure 7 shows that there is also only a very slight difference in the pattern for a crack and normal road.

Speed bumps are probably classified more accurately because they cause a much clearer pattern, which can be seen in figure 8.

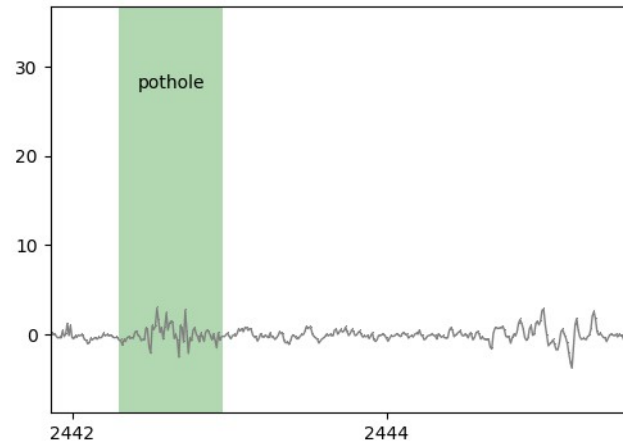


Figure 6. Vibration pattern of a pothole.

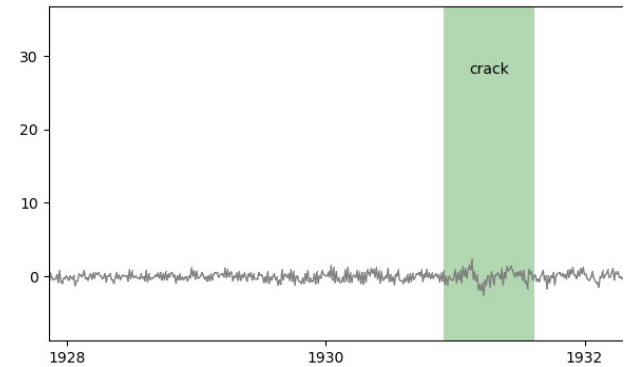


Figure 7. Vibration pattern of a crack.

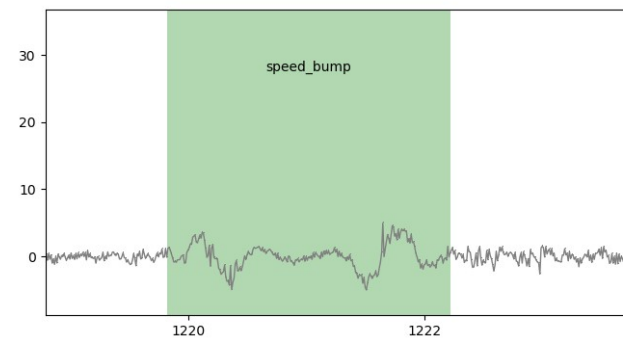


Figure 8. Vibration pattern of a speed bump.

Apart from the vibration patterns and the place of the sensor, it is also expected that the speed of the car will have an impact on the accuracy of the algorithms because when the car is driving slowly, the patterns will be different from when the car is driving faster. In this research, the speed of the car was not recorded while collecting the accelerometer data, and therefore the effect of speed was not researched.

Of the three places where the sensor was placed, the wrist seemed to be the worst position to be able to make correct classifications from. The most obvious reason for this seems that when the sensor is attached to the wrist, it will also pick up vibrations caused by the hand or arm of the driver. Figure 9 shows the vibration pattern that was caused by steering while the sensor was on the wrist.

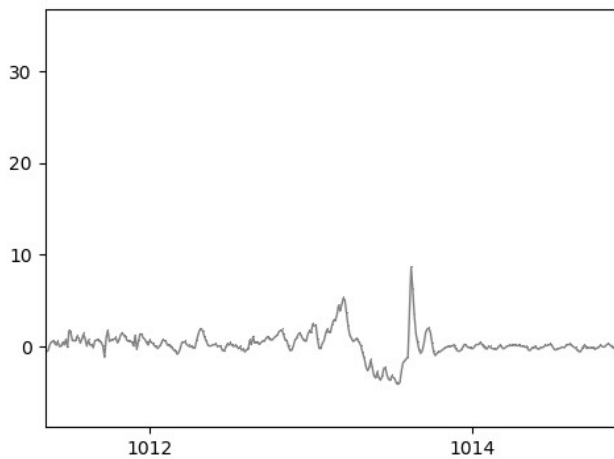


Figure 9. Vibration pattern of steering.

7. CONCLUSIONS

This research explored the feasibility of using a smartwatch and machine learning to classify road damage while driving. The results of the tests show that crocodile cracks and potholes could not be correctly classified using the methods discussed in this paper. The results of the tests also show that the position of the sensor affects the accuracy of the classifiers. When the sensor was placed on the wrist, which is where a smartwatch is normally worn, the accuracy of the classifiers was the worst. The reason that this position caused the worst results is assumed to be that when the sensor is on the wrist, it also picks up the movements that the driver makes with their arm and hand.

Of the used classifiers, the Naive Bayes classifier was slightly more accurate than the other classifiers, with an overall F_1 score of 0.303.

8. FURTHER RESEARCH

Further research could be done on testing with a real smartwatch, testing on different types of road pavement such as stones, testing the effects of the speed of the car, testing the effects of using different cars and on improving the pre-processing of the sensor-data such that the vibration patterns of the damaged parts of the road become clearer.

9. REFERENCES

- [1] F. S. Cabral, M. Pinto, F. A. L. N. Mouzinho, H. Fukai, and S. Tamura. An automatic survey system for paved and unpaved road classification and road anomaly detection using smartphone sensor. In *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, pages 65–70, July 2018.
- [2] M. Donlon. Pothole detection using embedded self-powered wireless sensors. <https://insights.globalspec.com/article/8344/pothole-detection-using-embedded-self-powered-wireless-sensors>, March 2018. [Accessed: December 2 2018].
- [3] M. Donlon. Using ai to locate potholes. <https://insights.globalspec.com/article/9517/using-ai-to-locate-potholes>, August 2018. Accessed: December 2 2018.
- [4] S. M. A. Gawad, A. E. Mougy, and M. A. El-Meligy. Dynamic mapping of road conditions using

smartphone sensors and machine learning techniques. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, Sept 2016.

- [5] Inertia-Technology. Promove mini. <http://inertia-technology.com/product/motion-capture-promove-mini/>. Accessed: January 24 2019.
- [6] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata. Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, 33(12):1127–1141, 2018.
- [7] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo. Real time pothole detection using android smartphones with accelerometers. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pages 1–6, June 2011.
- [8] pandas. Python data analysis library. <https://pandas.pydata.org/>. Accessed: January 27 2019.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [10] Scikit-learn. Scikit-learn algorithm cheat-sheet. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html. Accessed: November 12 2018.