APPLICATIONS OF MACHINE LEARNING FOR REAL-TIME ROAD ANOMALY IDENTIFICATION

An Undergraduate Research Scholars Thesis

by

DILLON KNOX

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Jim Ji

May 2018

Major: Electrical Engineering

TABLE OF CONTENTS

ABSTRACT

Applications of Machine Learning for Real-Time Road Anomaly Identification

Dillon Knox
Department of Electrical and Computer Engineering
Texas A&M University

Research Advisor: Dr. Jim Ji
Department of Electrical and Computer Engineering
Texas A&M University

Infrastructure degradation is becoming a wide-reaching problem in the United States, and there is a need to determine ways to intelligently distribute taxpayer money when addressing the issues. This paper investigates the use of smartphones to classify various road anomalies by using on-board sensors, including accelerometers, gyroscopes, and a cameras. Having a relatively robust sensor array in a ubiquitous device allows for crowdsourcing of data collection, and makes mapping large road networks that are prevalent in the US much more feasible. Specifically, this paper will propose a novel machine learning algorithm that can identify and differentiate between four different classifications of road anomalies, as opposed to the binary approach (using thresholding) that has been employed in similar studies. Additionally, this approach will be able to classify anomalies by severity, as well as provide an estimate of overall road roughness using the International Roughness Index (IRI). This data will allow for more accurate evaluations of overall road conditions than similar methods, and will allow preventive maintenance to be performed, potentially saving time and money.

ACKNOWLEDGEMENTS

I would like to thank my faculty advisor, Dr. Jim Ji, for first introducing me to research as an undergraduate and for guiding me on this project. I would also like to thank Akanksh Basavaraju for allowing to join and contribute to his research project, and for providing me with valuable insight throughout the entire process.

CHAPTER I

INTRODUCTION

Project Overview

Broadly speaking, the scope of this investigation can be subdivided into two stages: data collection and filtering (pre-processing), and algorithm development and analysis (post-processing). The first stage deals with the actual means of collecting data (via smartphone, mounted camera, stand-alone accelerometer), the quality of data collected, and approaches to clean up the data. The second stage involves the development of machine learning algorithm, including comparing and contrasting differing approaches (Support Vector Machine, k-nearest neighbors, and decision trees), initial training of each algorithm, and finally the results of applying each algorithm to our data set.

Pre-processing

In order to train an algorithm to differentiate between multiple types of road anomalies using just vibration (accelerometer) data, accelerometer data needs to be collected from a large quantity from each type of anomaly, so that the algorithm can identify the similarities between occurrences of the same anomaly. This requires collecting data from anomalies that have already been identified as falling into one of the pre-established categories (potholes, rutting, cracking, etc.). The most straight-forward way to do this is to collect video of each anomaly in order to be able to classify it after the fact. From the captured video, anomalies can be classified using an image processing algorithm that provides information about width, depth, and frequency of occurrence. Once an anomaly has been classified visually, its corresponding accelerometer data can be used to train the machine learning algorithm.

However, the raw collected data is undoubtedly noisy and need to be filtered. We propose a simple model where the raw data is first passed into an adaptive filter, in this case a Kalman filter, which will smooth the signal and eliminate noise, but will also preserve the high frequency and high amplitude signals that characterize a road anomaly. Next, the envelope of the filtered data will be extracted to produce a continuous function. Finally, the data will be windowed before post-processing begins.

Post-processing

Many previous studies have used a simple thresholding algorithm in order to classify anomalies in a binary fashion, i.e. an anomaly that causes a large enough spike in magnitude of the accelerometer s classified as a pothole; otherwise it is classified as not a pothole. We propose using machine learning to improve not only the predictive accuracy of our algorithm, but also the number of classifications that are possible.

We propose evaluating the performance of four different machine learning algorithms, which have traits ranging from ease of implementation to robustness of classification. The first algorithm that we will is decision tree based. Decision tree-based algorithms are comparatively simple and provide a baseline to compare the accuracy of other algorithms against. Developing the algorithm consists of determining several thresholds, with each threshold branching off to different thresholds depending on how a given input compares to it. It is important to note that the first test has the most impact on how the input is classified; for instance, a threshold for the accelerometer magnitude makes for a good first test. The k-nearest neighbors algorithm is slightly more robust than the decision tree-based algorithm, but is still simple to implement. The algorithm works by first graphing each point from the data set according to a set of given parameters that form the axes of the graph. The input in question is then also graphed and the distance to each data point from the training set is determined. The closest k number of data points is then sampled and the input is then classified as the classify that appears the most. The third and final algorithm that we will test is a Support Vector Machine (SVM), by far the most complex of the algorithms that we will test. Here, the algorithm is trained by graphing the training set on an n-dimensional space, where n is the given number of features that define the set. A hyperplane that separates the different classifiers is then determined by maximizing the separation between the plane and the given classifiers.

It should be noted that neural networks were not included a potential algorithm, despite being known for their high degree of accuracy, due to the large amount of training data required (Lépine).

After we have trained each algorithm, we will then test their accuracy by collecting more examples of anomalies, in a similar fashion to our training set, and running the algorithms, noting the true positive, false positive, and false negative rates for each. Based on these results, as well as additionally parameters such as runtime, we will be able to make a recommendation on what algorithm to use for future research.

CHAPTER II

METHODS

Data Acquisition

In order to identify and characterize road anomalies, three different sets of data are required: accelerometer data, video, and GPS data. Accelerometer and GPS data were collected using an iOS application developed by Akanksh Basavaraju, from the Department of Electrical and Computer Engineering at Texas A&M University. An iPhone 6s was used to run the app during data collection. Video was captured using a DJI Osmo camera that was provided by Dr. Eric Du, from the Department of Construction Science at Texas A&M University. The iPhone accelerometer has a frequency of 100 Hz, while the Osmo is capable of capturing 4K video, but was set at 720p due to storage limitations. A Ford Focus was used as the vehicle for data collection for the duration of the study. The iPhone was mounted at on the interior of the vehicle using a windshield mount, and the Osmo was mounted on the hood of the vehicle using a car mount, as shown in Figure 1. The iPhone was oriented at approximately a ninety degree angle, and any discrepancy was adjusted for using axis reorientation (see *Filtering* below). In this orientation, the y-axis represents the vertical motion of the vehicle, the x-axis represents horizontal motion, and the z-axis represents forward accelerations and decelerations.

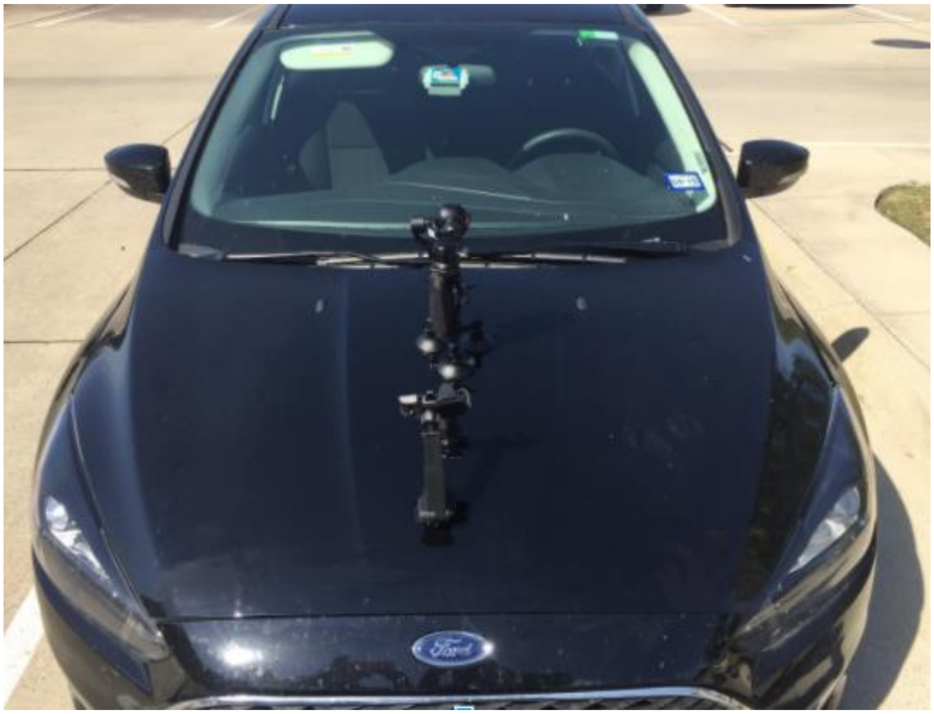

Figure 1. Positioning of the Osmo camera

Data was collected from roads around the Bryan-College Station metro area. Accelerometer, video, and GPS data was continuously recorded during stretches of road with known anomalies, and each stretch of road was ran over multiple times. The collected data was then hand-labeled in order to be used as a training set for the machine learning algorithms. The accelerometer data was labeled by inspecting the video from the Osmo and noting timestamps where anomalies approximately occur. Two different classes of anomalies, cracks and potholes were noted. Figure 2, below, shows an anomaly that would be labeled as a pothole through visual inspection, while Figure 3 shows an anomaly that would be labeled as a crack.

Figure 2. A frame shot with the Osmo showing a pothole

Figure 3. A frame shot with the Osmo showing a crack

The total accelerometer data was then sequentially windowed with a window length of 1 second, giving each window a total of 100 data points. If a window contained an anomaly, it was labeled as such, and windows that contained no visible anomalies were labeled as smooth road. Figure 4, below, shows the y-axis accelerometer data for a window that was labelled as containing a pothole. In total, 143 windows were labeled as containing potholes, 26 were labeled as containing cracks, and 817 were labeled as smooth road.

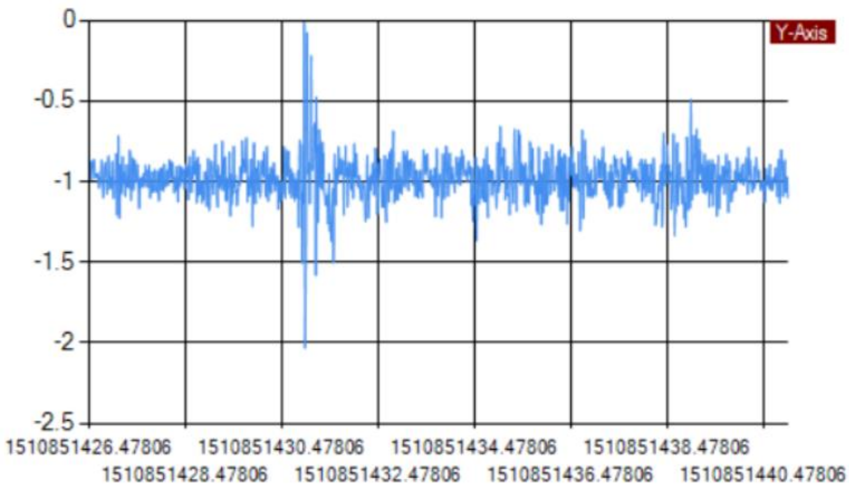

Figure 4. Windowed accelerometer data

Filtering and Feature Extraction

Even though the orientation of the smartphone was controlled throughout our initial study, it is important to consider that the orientation of each smartphone would not be controlled during any potentially crowdsourcing applications. Accordingly, we used the quaternion-based approach proposed by Tundo et al. to reorient the signal from all three axes so that negative y-axis of the accelerometer signal aligned with the gravitational acceleration vector (Tundo). After the raw accelerometer data was reoriented, the x and z axes were high-pass filtered. The high-pass filter removes low-frequency components from the accelerometer signal, and will thus remove the unwanted effects of turning, breaking, and acceleration (Eriksson). Finally, time-domain, frequency-domain, and wavelet features were calculated for each observation, and the results were subsequently organized into a matrix where each column represents a unique feature while each row represents a unique observation. Windowing, reorientation, filtering, and feature extraction were all performed using MATLAB.

Machine Learning Algorithms

Three different algorithms, k-nearest neighbors, Support Vector Machine, and a decision tree-based algorithm, were implemented using R. First, the feature data matrix containing 986 observations was imported from MATLAB. Before training the three algorithms, it was important to reduce the feature set in order to avoid potential overfitting, as well as to reduce training and classification time. First, a pairwise correlation matrix was produced, and highly correlated features were identified. If the correlation between two features was above 0.9, one of the features in the pair was declared redundant and subsequently discarded. Next, recursive feature elimination was used to determine how many features are necessary for classification, and which ones are the most important. A Support Vector Machine algorithm with a radial kernel was used to evaluate the performance of the model, first using all variables, and then removing the least important feature each iteration. Since the observation data contained an imbalanced number of classes (in particular, the number of smooth road observations dominated), log loss was chosen as the metric to minimize when evaluating all possible subsets of the features. Figure 5 shows the log loss of the model versus the number of features used. Log loss is minimized at 10 features, and plateaus thereafter.

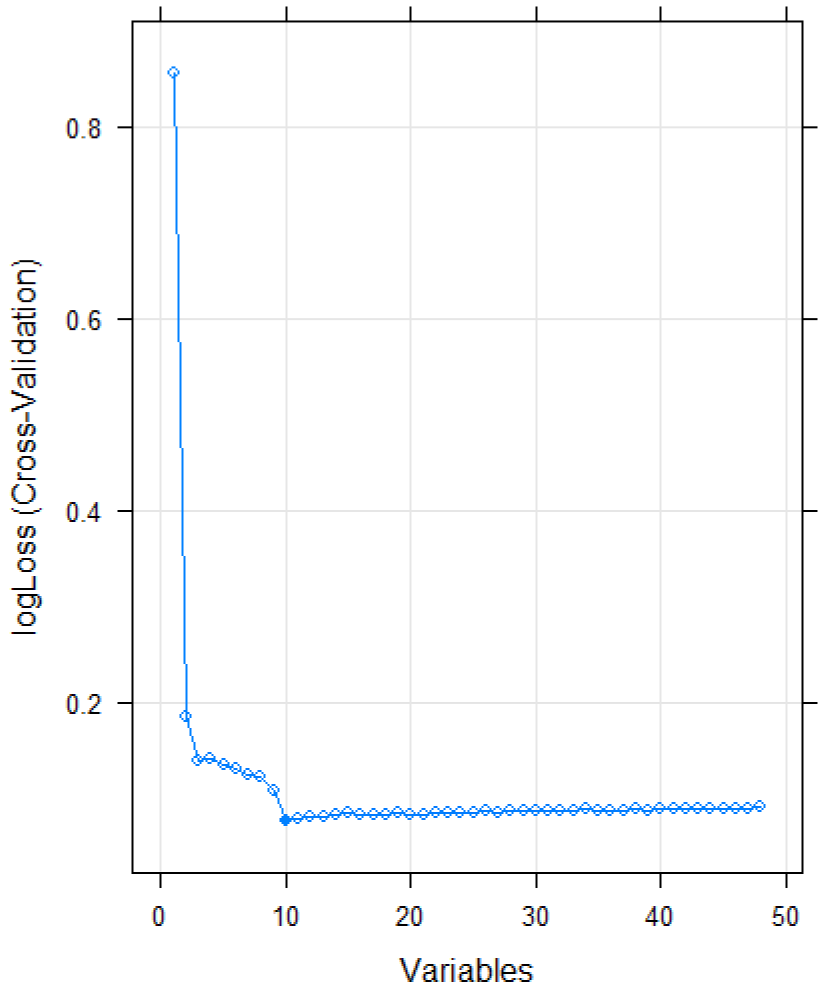

Figure 5. Log loss versus the number of features

The rfe function allows provides a ranking of the most important features, shown in Figure 6 below. The vast majority of the most important features involve power spectral density in some capacity.

Rank	Feature	Axis
1	PSD_AvgBandPower_5to15	Z
2	MeanMedianDiff_EnvUpper_LPwindow	X
3	PSD_AvgBandPower_30to40	Z
4	PSD_RMSBandValue_35to45	Y
5	PSD_MaxBandValue_30to40	Z
6	PSD_AvgBandPower_35to45	Y
7	PSD_MaxBandValue_5to15	Z
8	PSD_RMSBandValue_0to10	Y
9	PSD_AvgBandPower_5to15	X
10	PSD_AvgBandPower_40to50	X

Figure 6. Top 10 selected features

After the top features were selected, with data was randomly split into a training and testing set, with 75 percent used for training, and the remaining 25 percent used for testing. 10-fold cross validation was performed with each of the three models, and the resulting precision, recall, and F_1 scores were calculated using the testing data. Precision, recall, and F_1 score are mathematically calculated as follow:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

Additionally, training was repeated 10 times and the mean performance was calculated for each model. After an initial run to determine a baseline, further tuning was performed on each algorithm. First, the decision tree was pruned using the built-in *prune* function. Next, the k-nearest neighbors algorithm was optimized for the k-value that produced the highest accuracy

Finally, the SVM was optimized for the soft margin and gamma values (with each combination checked by cross-validation), as well as the kernel type that produced the highest F_1 score.

CHAPTER III

RESULTS

Algorithm Performance

Initially, the algorithms were trained and tested using a 75/25 split between training and testing data. The results of testing each algorithm are summarized in Table 1 below.

Table 1. Performance measures for each algorithm with a 75/25 testing/training split

Algorithm	Precision			Recall			F_1 Score		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.14	0.83	0.99	1.00	0.88	0.96	0.25	0.86	0.98
kNN	0.000	0.94	0.99	NaN	0.83	0.99	NaN	0.88	0.99
SVM	0.43	0.97	0.99	0.75	0.88	1.00	0.55	0.92	0.99

The SVM algorithm (with polynomial kernel) performed the best across all performance measures. Note that each algorithm performed exceptionally well at classifying smooth road and potholes, but struggled with classifying cracks. In particular, the precision of the SVM for cracks was well below what would be desirable, meaning that the algorithm struggles to positively predict cracks. However, the absolute number of observations for cracks was very low, and removing any observations from the training set may be extremely detrimental to performance. Therefore, it is useful to perform training with all the observations, keeping in mind the potential risk of overfitting to the training set when evaluating each algorithm's performance. Each algorithm was trained and tested again, this time using all observations for testing data, and the precision, recall and F_1 score were calculated (see Table 2, below).

Table 2. Performance measures for each algorithm with all data used for training

Algorithm	Precision			Recall			F_1 Score		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.42	1.00	0.99	0.92	0.88	0.99	0.58	0.94	0.99
kNN	0.54	0.96	0.99	0.70	0.90	0.99	0.61	0.93	0.99
SVM	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

As expected, the performance of each algorithm improves immensely, especially in terms of being able to classifying cracks. Note, however, that the SVM (now maximized by a radial kernel) achieves perfect classification. This is likely a sign of overfitting; nevertheless, the fact that all three algorithms saw significant improvements across all statistical measures when classifying cracks shows that there is potentially a great benefit to collecting more data for training purposes.

Comparison to Previous Research

In recent years, a lot of research has been conducted on classifying road anomalies in an efficient and crowd-sourceable fashion. Relevant previous research is summarized below.

Many studies have proposed using a simple thresholding approach on accelerometer data in order to classify potholes and smooth road. Kulkarni et al. were able to achieve pothole detection accuracy between 90 and 95% by simply determining threshold values on the x and z axes (Kulkarni). Tai et al. also used thresholding to detect general road anomalies, and were able to achieve a precision of 0.785 and a recall of 0.705% (Tai). Outside of using accelerometer data, Tedeschi and Bendetto proposed an image processing-based solution, where users of an Android app are able to take pictures of a roadway, and an algorithm detects whether a pothole or crack is present. For cracks, this approach was able to achieve a precision of 0.767, a recall of 0.736, and an F_1 Score of 0.769. For potholes, it was able to achieve a precision of 0.812, a recall of 0.767, and an F_1 Score of 0.792 (Tedeschi). Finally, Allouch et al. were able to develop a smartphone application, RoadSense, that used both accelerometer and gyroscope data to classify potholes and smooth road, and were able to achieve a precision of 0.951, a recall of 0.953, and an F_1 Score of 0.950 using SVM (Allouch).

Our results compare very favorably to the results of previous studies. Our precision of 0.972 when classifying potholes is higher than comparable studies, and our F_1 Score of 0.921 is comparable to the results obtained using RoadSense. Additionally, our approach is non-binary, allowing for cracks and potholes to be classified, in addition to smooth road. Finally, our algorithm only depends on accelerometer data, and only uses 10 features for classification, potentially allowing for fast classification times in real-time applications.

CHAPTER IV

CONCLUSION

In this paper, a machine-learning approach to detecting road anomalies was proposed that uses accelerometer data collected using a smartphone. Three different classification algorithms were tested on classifying two types of anomalies (cracks and potholes) and smooth road. SVM was determined to be superior at classifying each of the three classes across multiple statistical measures. The performance of our algorithm was much better than previous studies that used thresholding to classifying potholes and smooth road, with precision and recall values up to 24 percent higher. When compared to similarly implemented SVM algorithms that used both accelerometer and gyroscope data, our best algorithm was able to achieve a higher precision and similar F_1 score (0.92 vs 0.95), while only using accelerometer data. Additionally, it was also able to differentiate between cracks and potholes and potentially offers very fast runtimes for real-time classification, because only 10 total features are used.

Future work will be focused on collecting more data for algorithm training purposes, as the performance of our algorithm in terms of cracks was potentially limited by the comparatively small number of observations. Additional proposals for future work include incorporating GPS data into a mobile application that can be used for crowdsourcing, and using the collected accelerometer data to provide a measure of road roughness in addition to identifying anomalies.

REFERENCES

Allouch, Azza, et al. “RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope.” *IEEE Sensors Journal*, vol. 17, no. 13, Jan. 2017, pp. 4231–4238.

Eriksson, Jakob, et al. “The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring.” 2008.

Kulkarni, Aniket, et al. “Pothole Detection System Using Machine Learning on Android.” *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, no. 7, July 2014.

Lépine, Julien. “On the Use of Machine Learning to Detect Shocks in Road Vehicle Vibration Signals.” Victoria University, 2016.

Tai, Yu-chin, et al. “Automatic Road Anomaly Detection Using Smart Mobile Device.” National Taiwan University.

Tedeschi, A., and F. Benedetto. “A Real-Time Automatic Pavement Crack and Pothole Recognition System for Mobile Android-Based Devices.” *Advanced Engineering Informatics*, vol. 32, 2017, pp. 11–25.

Tundo, Marco, et al. “Correcting Smartphone Orientation for Accelerometer-Based Analysis.” University of Ottawa.