

# MONITORING ROAD SURFACE CONDITIONS WITH CYCLIST'S SMARTPHONE SENSORS

Budi Darma Setiawan<sup>1,2</sup>, Victor V. Kryssanov<sup>1</sup>, Uwe Serdült<sup>1,3</sup>

<sup>1</sup> Graduate School of Information Science and Engineering, Ritsumeikan University, Japan

<sup>2</sup> Faculty of Computer Science, Universitas Brawijaya, Indonesia, s.budidarma@ub.ac.id

<sup>3</sup> Center for Democracy Studies Aarau (ZDA), University of Zurich, Switzerland

## ABSTRACT

Road networks form one of the most important infrastructures in modern cities, while road conditions determine the very possibility and quality of land transportation. It is therefore important to monitor and manage road networks properly. The vast area that should be monitored and managed makes this task both expensive and time-consuming. Recently, an approach to involve road users, such as car drivers, pedestrians, and cyclists, to participate in monitoring road conditions has emerged. Monitoring roads using bicycles has an advantage, compared to using a car, since it allows for reaching narrow roads. This paper presents results of a preliminary study of using a bicycle for detecting road surface defects including potholes, and bumps. Data collected with a cyclist's smartphone sensors was used to train artificial neural networks in different configurations. The trained networks were then used to detect road surface defects. Results obtained in the experiments indicate that for the accelerometer data, a convolutional neural network provides for the best average accuracy in classifying road surface conditions. Also, this and a long short term memory network produce better results than a standard deep neural network.

**Keywords:** Road condition monitoring, smartphone applications, artificial neural networks.

## 1. INTRODUCTION

The very possibility of land transportation depends on road conditions as a poor quality of the road would fatally disrupt the traffic flow. It is, therefore, important to maintain roads thoroughly and on a regular basis. Due to the enormous size of road networks in modern cities, monitoring road conditions is a time-consuming and expensive task. One approach to cope with this problem is to get road users, such as pedestrians, car drivers, and cyclists, involved in the monitoring process.

Recently, several smartphone applications have been developed to monitor road conditions (Allouch *et al.*, 2017; Li and Goldberg, 2018; Mednis *et al.*, 2011; Varona *et al.*, 2019). The focus of the reported studies was on using smartphones placed inside a car to automatically detect road surface conditions while the car is driven. Naturally, however, a car can only be used for monitoring sufficiently wide roads. The focus of the presented work is on using a bicycle for the same purpose.

Smartphones are convenient to use for road monitoring, as they are equipped with GPS sensors for tracking locations of road defects. There are also several movement sensors, such as accelerometer, gyroscope, and magnetometer. The idea underlying this study is that when the smartphone is carried by the cyclist through a pothole or defect on the road, the sensors register vibrations, and the data would be used to detect the road surface defect. As not all vibrations are caused by road defects, human-made structures, such as speed bumps, would wrongly be detected as defects. It is, therefore, important to develop a method allowing for reliable classification of road surface conditions that would ignore road structures not requiring maintenance.

Potholes and bumps can be recognized by analyzing patterns of a signal generated by accelerometer or gyroscope sensors. There have been studies reported that used machine learning techniques, such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short Term Memory (LSTM), to achieve this goal (Hur *et al.*, 2018; Hussain *et al.*, 2019; Lee *et al.*, 2017). This presented study attempts to find the most efficient method when dealing with signals registered with cyclist smartphones.

## 2. RELATED WORK

Results of a study on road condition monitoring using bicycles have recently been reported (Werner, 2018). The author evaluated the capability of smartphone sensors to register vibrations generated when riding a bicycle, and used vibration data collected to evaluate the quality of bicycle tracks. The assessment was done by calculating the Dynamic Comfort Index (DCI) that reflects the comfort of riding a bicycle. Three types of smartphone sensors

were used in the study: accelerometer, linear accelerometer, and GPS receiver. While a detailed analysis of various smartphone vibrations affecting DCI values has been made, the study mainly focused on the rider's comfort rather than on the road surface conditions.

Li and Goldberg (2018) conducted research on using smartphone accelerometers to evaluate the overall condition of motor-vehicle roads. The study, however, did not attempt to differentiate road defects by type.

Varona *et al.* (2019) tried to combine previous studies on road surface monitoring and road surface material as well as pothole and bump detection. The author also used smartphones installed inside a car. Several machine learning techniques, namely convolutional neural networks, long short-term memory, and Reservoir Computing (RC) were used in the study. The authors claimed that the use of machine learning can improve the real-world scenario of detecting potholes and man-made structures (e.g. speed bumps). Input features for the developed system include accelerometer coordinate values ( $x$ ,  $y$ , and  $z$ ) and differences of the values notated as  $diffX$ ,  $diffY$ , and  $diffZ$ . The same features are used in the presented study, together with CNN and LSTM.

Accelerometer data is frequently used for road condition monitoring, whether it comes from a smartphone, an accelerometer sensor embedded into a micro-computer, such as Raspberry Pi, or from an accelerometer installed in a vehicle (Allouch *et al.*, 2017; Devekar *et al.*, 2018; Park *et al.*, 2018). Other sensors, such as gyroscope, steering angle, and wheel speed sensors were also used in the related studies. Installing and configuring steering angle and wheel speed sensors in the case of a bicycle is a complex task, yet not all bicycles would support such installations. Hence the presented study only deals with sensors generally available in a smartphone, i.e. with accelerometers and gyroscopes.

### 3. DATA

#### 3.1 Data Collection

An Android application has been developed for data collection. The application records sequential values read from the accelerometer and gyroscope sensors, but also locations from the GPS. The recording frequency is 50 Hz. Figure 1(a) presents the application interface. The accelerometer is used since it measures the acceleration applied to the phone, while the gyroscope measures the orientation change of the phone that might be experienced by the phone when a cyclist rides a bicycle through holes and bumps.

The developed application has been installed on a smartphone, and the phone was put in a cyclist's shirt pocket so that it could record vibrations while the cyclist was riding a bicycle. The data captured with the application was saved in CSV-formatted files.

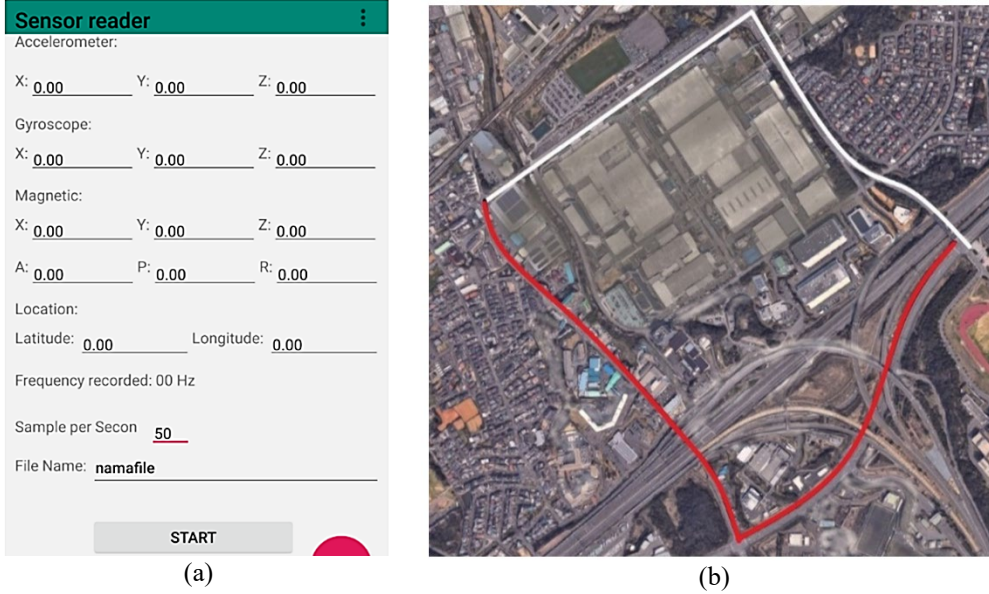
Data was collected while driving on two road lanes as shown in Figure 1(b). The paths included several speedbumps and road defects that were used for training and testing machine learning algorithms. In Figure 1(b), the white line indicates the path used to collect data for the training purposes, and the red line indicates the path used for testing.

#### 3.2 Data Preprocessing

The collected data was preprocessed by slicing it into several chunks. A sliding window was used to create the chunks. For the window size, Mednis *et al.* (2011) got a maximal true positive at 20 samples on the 100Hz sampling rate and 0.2 seconds window size, while Varona *et al.* (2019) used 85 samples on the 50Hz sampling rate and 1.8 seconds window length. In the presented study, the window length was set at 25 samples on the 50 Hz sampling rate and 0.5 seconds on window length. This was done, based on assumptions that for a bicycle average speed, the time required to drive through an obstacle (pothole or bump) is approximately 0.5 seconds. The window is shifted by 10 values to capture the next chunk. The process is repeated until all the data is arranged into chunks.

In the preprocessing step, every chunk got four features: original accelerometer data defined as  $acc \in \{accX, accY, accZ\}$ , original gyroscope data defined as  $gyr \in \{gyrX, gyrY, gyrZ\}$ , numerical differential of the accelerometer data defined as  $diffAcc \in \{diffAccX, diffAccY, diffAccZ\}$ , and numerical differential of the gyroscope data defined as  $diffGyr \in \{diffGyrX, diffGyrY, diffGyrZ\}$ . All  $x$ ,  $y$ , and  $z$  notations refer to the three-dimensional space coordinates.

Two features,  $diffAcc$  and  $diffGyr$ , which are numerical differentials of the original accelerometer and gyroscope data vectors, respectively, were obtained using equations (1) and (2). In this case,  $i$  is the sequential value number in a chunk, " $acc$ " signifies accelerometer data, and " $gyr$ " is for gyroscope data.



**Fig 1. (a) User interface of the developed Android sensor reader application; (b) Paths used to collect the data (Map data: Google, CNES/Airbus, Maxar Technologies)**

$$\begin{aligned}
 \text{diffAccX}(i) &= \text{accX}(i+1) - \text{accX}(i) \\
 \text{diffAccY}(i) &= \text{accY}(i+1) - \text{accY}(i) \\
 \text{diffAccZ}(i) &= \text{accZ}(i+1) - \text{accZ}(i)
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 \text{diffGyrX}(i) &= \text{gyrX}(i+1) - \text{gyrX}(i) \\
 \text{diffGyrY}(i) &= \text{gyrY}(i+1) - \text{gyrY}(i) \\
 \text{diffGyrZ}(i) &= \text{gyrZ}(i+1) - \text{gyrZ}(i)
 \end{aligned} \tag{2}$$

Each chunk preprocessed as described was labeled manually by assigning one of the three labels: “normal”, “pothole”, or “bump”. For training and evaluation purposes, 359 chunks were selected randomly, 309 chunks were used for training (125 chunks labeled as “normal”, 92 as “pothole”, and 92 as “bump”), and the remaining 50 chunks were used for validation.

## 4. METHOD

Three types of artificial neural networks were tested: DNN, CNN, and LSTM. In each case, the input is a chunk of one of the following: *acc*, *gyr*, *diffAcc*, or *diffGyr*. As each chunk is a 3-dimensional vector (in x, y, and z coordinates), the arrangement of the data is different for each neural network architecture.

### 4.1 Deep Neural Network

The DNN implemented in this study has a simple neural network architecture with 4 hidden layers, each of which is a fully connected layer of 150 hidden units (artificial neurons). A drop-out layer for each of the hidden layers was implemented to prevent the network from overfitting (Srivastava *et al.*, 2014). The drop-out probability was set at 0.2.

For the input, each chunk is rearranged sequentially in the order of the 3 coordinates (x, y, and z), so the input length of *acc* and *gyr* becomes 75, formed by 3 x 25 coordinate values from the chunk. Since *diffAcc* and *diffGyr* are calculated by subtracting the  $i^{\text{th}}$  values from the  $(i-1)^{\text{th}}$  values, the length of the corresponding chunks is 72 (formed by 3 x 24 differential coordinate values).

For the output layer, the DNN used has 3 output units representing the road surface condition classes (“normal”, “pothole”, and “bump”).

This architecture uses the categorical cross-entropy loss function and the Adam optimizer (learning rate = 0.001, beta1 = 0.9, and beta2 = 0.999). The activation function implemented for each hidden unit is rectified linear, while softmax is implemented for the output unit.

## 4.2 Convolutional Neural Network

In the implemented CNN, input sequences of chunks are rearranged into a two-dimensional array. First, the x, y, and z sequences are arranged in a one-dimensional sequential array in the order of x, y, and z, respectively. Next, the obtained array is reshaped into a two-dimensional array. For example, the *acc* and *gyr* features, each is represented as a one-dimensional array of the length of 75, then re-represented in a two-dimensional array of 5 x 15. Likewise, the *diffAcc* and *diffGyr* features are arranged in a one-dimensional array of the length of 72 each and then re-represented as a two-dimensional array of the 8 x 9 size each.

Two convolutional layers are used, equipped with maxpooling of the 2 x 2 size. The first convolutional layer uses 56 kernels (3 x 3 size), and the second layer uses 128 kernels (also 3 x 3). There is also a flattening fully connected layer of 150 hidden units. Finally, there is one output layer with 3 output units representing the output classes. Figure 2 depicts the architecture of the CNN used.

The implemented CNN also uses categorical cross entropy as the loss function, and Adam as the optimizer with learning rate = 0.001, beta1 = 0.9, and beta2 = 0.999. Each convolutional layer uses rectified linear as the activation function, and each output unit uses softmax.

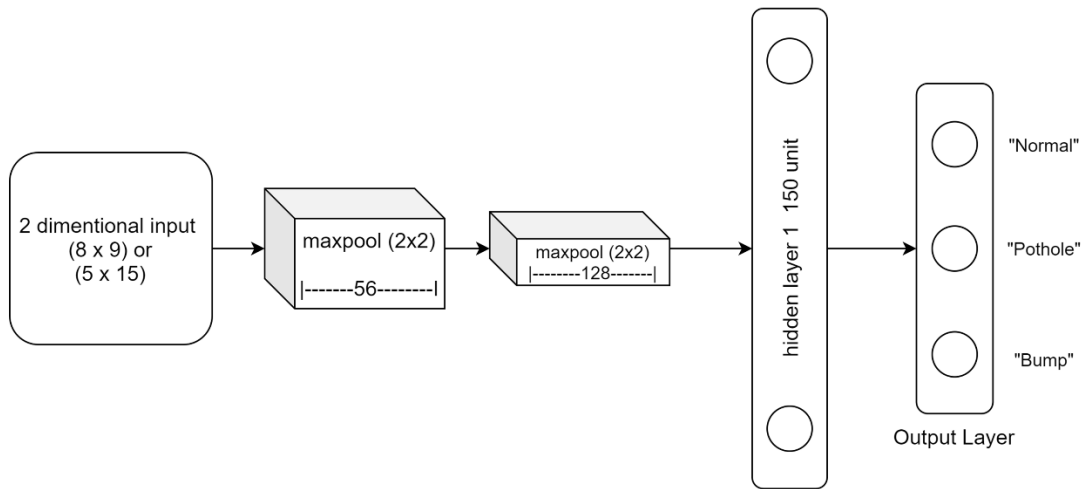


Fig 2. CNN Architecture

## 4.3 Long Short Term Memory

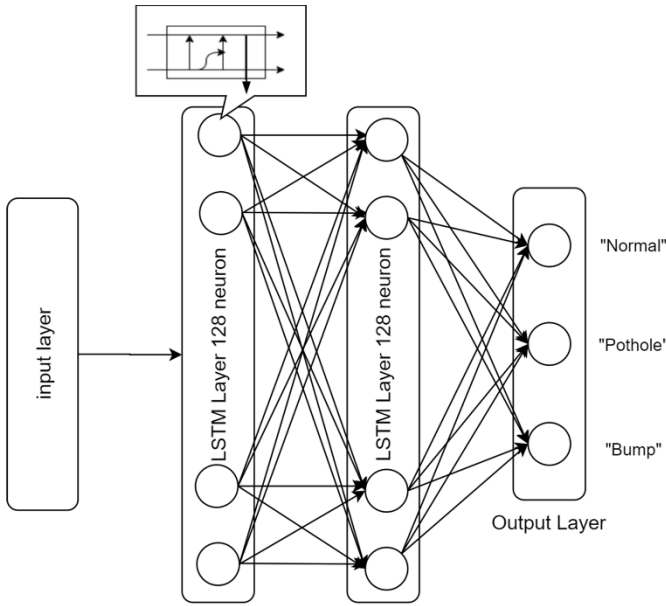
LSTM is often used to analyze sequential data, and to make time-series predictions (Hussain *et al.*, 2019). Since the recorded sensor data is time series, an attempt was made to classify the sequence of accelerometer data using an LSTM, as originally proposed in Varona *et al.* (2019).

This proposed study uses two LSTM layers, each consisting of 128 hidden units. The input is defined as one chunk of a length of 75 or 72 values, which is a sequential arrangement of x, y, and z values. One value of the chunk is processed at one time-step. The output layer is a dens layer which has 3 units corresponding to the class labels. The LSTM architecture is shown in Figure 3.

The LSTM is configured, using the categorical cross-entropy loss function and the Adam optimizer (learning rate = 0.001, beta1 = 0.9, and beta2 = 0.999).

## 5. RESULTS

Each of the three artificial neural networks was trained and tested five times, assessing the average and the best accuracy achieved. Figure 4 presents results obtained for the four different features. The maximum accuracy of 93.88% was obtained when using the CNN for the *diffACC* and *diffGyr* features, and the LSTM for the *acc* and *diffAcc* features. The best average accuracy of 91.84% was registered when the CNN was used with the *diffAcc* data.

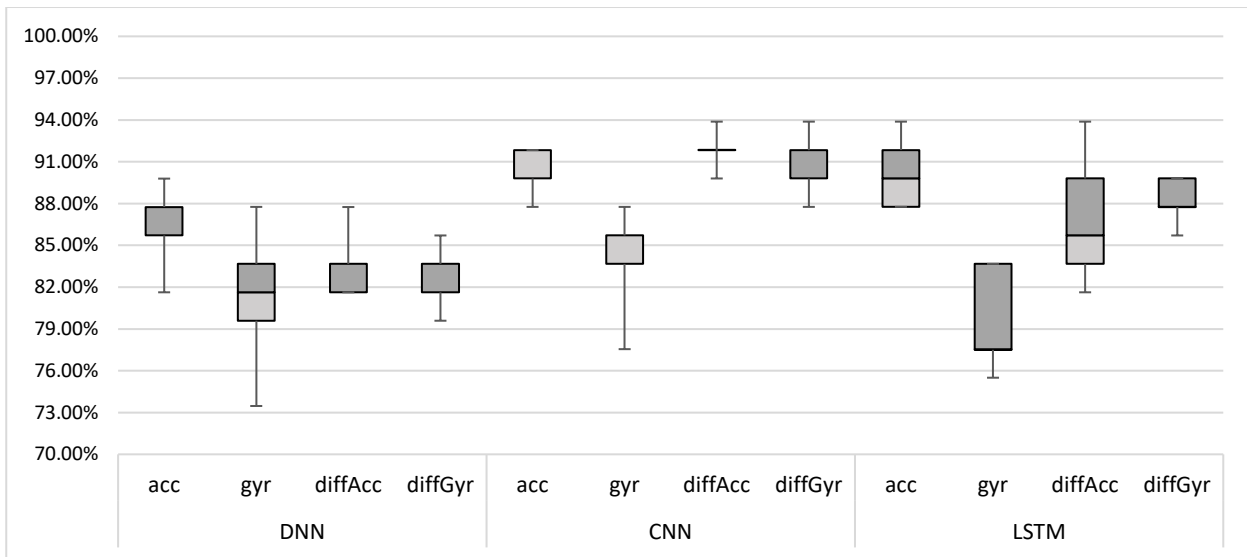


**Fig 3. LSTM Architecture**

As it can be seen from Figure 4, the classification using the gyroscope data always gets worse results (average accuracy) and stability (assessed with the variance) compared to when using the accelerometer data. It shows that for this case of classification, accelerometer is a better feature compared to the gyroscope. This could be caused by the fact that the gyroscope is more sensitive to orientation change than to acceleration, and vibration patterns due to driving through potholes and bumps reflect not orientation but acceleration changes.

The experimental results also showed that the training process is more stable when using the CNN and the *diffAcc* data. It can be seen in Figure 4 that by using the CNN and the *diffAcc* data, the maximum and minimum accuracies are not very different. It can also be seen that the CNN and the LSTM produced better results than the DNN, except for when using the original gyroscope data (*gyr*).

The corresponding confusion matrices are given in Figure 5 that shows only the matrices for the four combinations with the best maximum accuracy (CNN for the *diffAcc* data, CNN for *diffGyr*, LSTM for *acc*, and LSTM for *diffAcc*). As one can see, most of the false-negatives are associated with classifying potholes (for example, in matrix (a), 25% of the potholes cases were recognized as bumps). The “normal” road surface classified correctly using the CNN and *diffAcc*, whereas the bumps are best recognized when using the LSTM with both *acc* and *diffAcc*.



**Fig 4. Accuracy comparison for the DNN, CNN, and LSTM**

	c0	c1	c2
c0	1	0	0
c1	0	.75	.25
c2	0	0	1

(a)

	c0	c1	c2
c0	1	0	0
c1	.08	.75	.17
c2	0	0	1

(b)

	c0	c1	c2
c0	1	0	0
c1	.25	.75	0
c2	0	0	1

(c)

	c0	c1	c2
c0	.95	.05	0
c1	.17	.83	0
c2	0	0	1

(d)

c0: class "normal"  
c1: class "pothole"  
c2: class "bump"

**Fig 5. Confusion matrices of (a) CNN for the *diffAcc*, (b) CNN for the *diffGyr*, (c) LSTM for the *acc*, and (d) LSTM for the *diffAcc***



**Fig 6. Real-world test results (Map data: Google, CNES/Airbus, Maxar Technologies)**

The best scenario (the CNN for the *diffAcc* data) was used in a real-world test. The red line in Figure 1(b) shows the test path, and test results are presented in Figure 6, where red spots indicate detected potholes and blue spots indicate detected bumps. Not all potholes and bumps shown in Figure 6 were recognized as expected. Some recognitions were due to the rough surface of the road. It happened since rough surface roads could be considered as roads with multiple potholes and bumps. Therefore, instead of detecting only potholes and speed bumps, one should also detect rough surface roads. In Figure 6, the dots not always appear in the exact position as it should be. As the smartphone GPS accuracy would not be sufficiently high, the detected road defects sometimes appear as far as 5 meters away from the real locations.

## 6. CONCLUSIONS

It has been confirmed in this study that it is possible to detect potholes and speed bumps using machine learning methods. It was shown that deploying CNN with *diffAcc* data provides for the best detection accuracy. It was also found that accelerometer data fares better compared to gyroscope data.

By placing the smartphone inside a shirt pocket, relatively good results were achieved in spite of the possible driver movements that would affect classification results. While inside a shirt pocket, the phone still could sense vibrations caused by driving through potholes and bumps.

Additional work is required to investigate the influence of smartphone orientation and position, and also the rider's speed on the recognition of road surface defects with smartphone sensors. Different orientation and position of the smartphone would generate different signal pattern. The signal could also be affected by the driver's movements.

## REFERENCES

- Allouch, A., Koubaa, A., Abbes, T., and Ammar, A. (2017). RoadSense: Smartphone Application to Estimate Road Conditions Using Accelerometer and Gyroscope. *IEEE Sensors Journal*, 17(13), 4231–4238.
- Devekar, N., Damodar, S., Shendkar, P., Mulani, W., and Narde, V. (2018). Pothole Detection System for Monitoring Road & Traffic Conditions using IoT. *IJIRST - International Journal for Innovative Research in Science & Technology*, 5(7), 34–37.
- Google map. (n.d). Nojihigashi and Kasayama, Kusatsu, Shiga, Japan. <https://www.google.com/maps/place/525-0072,+Japan/@34.9846575,135.9517519,1646m/data=!3m1!1e3!4m5!3m4!1s0x60016d99c1b8cf85:0xe57167bf5db00a68!8m2!3d34.9796111!4d135.9503377> (last accessed on December 18, 2019)
- Hur, T., Bang, J., Huynh-The, T., Lee, J., Kim, J. I., and Lee, S. (2018). Iss2Image: A novel signal-encoding technique for CNN-based human activity recognition. *Sensors*, 18(11), 1–19.
- Hussain, G., Jabbar, M.S., Cho, J.D., and Bae, S. (2019). Indoor positioning system: A new approach based on lstm and two stage activity classification. *Electronics*, 8(4), 1–27.
- Lee, S.M., Yoon, S.M., and Cho, H. (2017). Human activity recognition from accelerometer data using Convolutional Neural Network. In: *Proceeding of International Conference on Big Data and Smart Computing, BigComp*, 131–134, IEEE: New York NY.
- Li, X., and Goldberg, D.W. (2018). Toward a mobile crowdsensing system for road surface assessment. *Computers, Environment and Urban Systems*, 69, 51–62.
- Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., and Selavo, L. (2011). Real time pothole detection using Android smartphones with accelerometers, In: *Proceeding of the International Conference on Distributed Computing in Sensor Systems and Workshops*, 1–6, IEEE: New York NY.
- Park, J., Min, K., Kim, H., Lee, W., Cho, G., and Huh, K. (2018). Road surface classification using a deep ensemble network with sensor feature selection. *Sensors*, 18(12), 1–16.
- Srivastava, N., Hinton, G., Krizhevsky, A., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- Varona, B., Monteserin, A., and Teyseyre, A. (2019). A deep learning approach to automatic road surface monitoring and pothole detection. *Personal and Ubiquitous Computing*. doi: [10.1007/s00779-019-01234-z](https://doi.org/10.1007/s00779-019-01234-z)
- Werner, S.P. (2018). *A monitoring system for bicycle pavement conditions using cyclists' smartphones a first look at the capabilities of measuring bicycle vibrations using smartphone sensors*, Master's Thesis, Eindhoven University of Technology, Netherlands.