Personal and Ubiquitous Computing
https://doi.org/10.1007/s00779-019-01234-z

ORIGINAL ARTICLE

A deep learning approach to automatic road surface monitoring and pothole detection

Braian Varona[1] · Ariel Monteserin[1] · Alfredo Teyseyre[1]

Received: 1 December 2018 / Accepted: 8 May 2019

Abstract

Anomalies in road surface not only impact road quality but also affect driver safety, mechanic structure of the vehicles, and fuel consumption. Several approaches have been proposed to automatic monitoring of the road surface condition in order to assess road roughness and to detect potholes. Some of these approaches adopt a crowdsensing perspective by using a built-in smartphone accelerometer to sense the road surface. Although the crowdsensing perspective has several advantages as ubiquitousness and low cost, it has certain sensibility to the false positives produced by man-made structures, driver actions, and road surface characteristics that cannot be considered as road anomalies. For this reason, we propose a deep learning approach that allows us (a) to automatically identify the different kinds of road surface, and (b) to automatically distinguish potholes from destabilizations produced by speed bumps or driver actions in the crowdsensing-based application context. In particular, we analyze and apply different deep learning models: convolutional neural networks, LSTM networks, and reservoir computing models. The experiments were carried out with real-world information, and the results showed a promising accuracy in solving both problems.

Keywords Deep learning · Road surface monitoring · Pothole detection · Crowdsensing

1 Introduction

Maintaining the road infrastructure free of potholes is a difficult task. Several works have been proposed to solve the problem of detecting potholes and other road defects by using mobile devices [2, 10, 27, 30, 33, 36, 42, 43, 48, 49]. All these works share the idea that the potholes can be detected by analyzing the acceleration values sensed by a mobile device. That is, when a mobile device is placed in a fixed position inside the vehicle, the device movements can reflect the movements of the vehicle. Then, the presence of potholes and its severity, in terms of destabilization of the vehicle, can be detected by analyzing the accelerometer's information.

However, not all destabilizations in a vehicle can be produced by road defects. The road infrastructure usually has man-made speed bumps and other structures that produce vibrations in a vehicle but cannot be considered road defects. This fact produces a large number of false positives, which degrades the accuracy of pothole detection [36]. Moreover, there are different kinds of road surface: concrete panels, cobblestones, asphalt, and dirt roads, among others. Usually, these kinds of road surface also destabilize the vehicle in different ways. For this reason, it is important to have information about the road surface through which the vehicle is traveling. For example, a destabilization produced in a vehicle while traveling on a cobblestone road could be considered a pothole if the same destabilization is detected in an asphalt road. In addition, the driver can do actions that affect the correct sensing of the road. For example, the driver could move the mobile device to answer a call or reply to a message.

In this work, we analyze different deep learning approaches to automatically detect those situations that affect the vehicle stability but cannot be considered as road defects. On the one hand, we aim to automatically identify

✉ Ariel Monteserin
ariel.monteserin@isistan.unicen.edu.ar

Braian Varona
braian.varona@isistan.unicen.edu.ar

Alfredo Teyseyre
alfredo.teyseyre@isistan.unicen.edu.ar

[1] ISISTAN Research Institute (CONICET-UNICEN), Tandil, Argentina

the different kinds of road surfaces. This is particularly useful if we want to make available a service of pothole detection in multiple cities, especially because there are usually no public information about road infrastructure. On the other hand, we aim to automatically distinguish destabilizations produced by a pothole from destabilizations produced by man-made speed bumps or driver actions.

First, a mobile application continually records the accelerometer oscillations and GPS information while the vehicle travels along a road. Thus, each track is represented by a sequence of oscillations produced by the accelerometers of the mobile device. An entry of a sequence is composed of a three pair of values (the raw value of the accelerometer and its difference with the previous value) for axis X, Y, and Z, and the respective information of geolocalization (latitude and longitude). In a single track, different stability events can be detected. We named stability event to the events that produce loss of stability in the vehicle. Each event is composed of a finite sequence of oscillations. These sequences include the oscillations since the event occurs (for example, a pothole is impacted or the driver answers a call) until the vehicle stabilizes (i.e., the oscillations disappear). We model these sequences as multivariate time series. A multivariate time series is a sequence of numerical vectors [46].

Several approaches to multivariate time series classification have been proposed. In this work, to automatically identify the different kinds of road surface, we train three deep learning models: a convolutional neural network (CNN), a long short-term memory (LSTM) neural network, and a reservoir computing (RC) model, using a set of sequences corresponding to concrete panels, cobblestones, asphalt, and dirt roads. Moreover, we train other instances of these models with different kinds of stability events. These events include destabilizations produced by potholes, speed humps, street gutters, and driver actions (answer a call, reply to a message, and open/close the vehicle door). To solve both problems, we propose several alternative techniques for data augmentation and segmentation. These techniques allow us to improve the precision of the models.

The approach was evaluated by using real-world information obtaining an accuracy of 85% for road surface classification and 93% for stability event classification. Both best results were obtained using the CNN model. In addition to the deep learning models, we compared the results with other baseline approaches. These approaches include a multivariate time series classification approach based on feature extraction [20], and dynamic time warping [40] and an approach based on clustering and classification for gesture recognition [18].

The rest of the article is organized as follows. Section 2 presents some background and related work on road roughness classification and automatic pothole detection and some related concepts. In Section 3, we describe the proposed approach to road surface classification and pothole detection. Section 4 presents the experimental evaluation and the results. Finally, Section 4 informs the conclusions and future work.

2 Road roughness classification and automatic pothole detection

Due to the high cost of keeping road surface in good conditions, automatic monitoring systems are very important to reduce accidents, improve traffic safety, and protect vehicles from damage. In the last years, several approaches have been proposed to solve the problem of classifying the road roughness and detecting potholes. The first approaches [2, 15, 21, 32, 47–50] focus on determining the roughness of different sections of a road according to a roughness index (usually, the International Roughness Index [1]). Roughness is a global measure of road quality, and it has long been a major criterion in the assessment of road conditions to guide resource allocation and maintenance planning [50]. Basically, these approaches classify a section within a roughness scale, for example, from “very good” (smooth road) to “very poor” (very rough road).

On the other hand, some approaches focus on specifically detecting potholes and road anomalies [5, 10, 12, 15, 23, 24, 27, 30, 33, 36, 42, 43]. In general, these approaches analyze the input data and determine the existence of a pothole/anomaly. However, most of these approaches just take into account two classes: for example, anomaly-normal [33]; smooth-road bump [30]; and shock-no shock [24]; among others. In fact, some works consider that multiple classes of bumps exist like speed bumps, manholes, potholes, rough patches, reflectors, and joints [36], but just classify events or not events. In particular, [16] presents a heuristic to differentiate between potholes and humps, but it is tested with one pothole and one hump. Other works extend the classification by the anomaly severity [15, 42]. Nevertheless, none of the approaches have considered a mechanism to distinguish potholes from speed bumps or driver actions, classifying the last also as road anomalies. Indeed, some works discard the road sections with highway reflectors, due to the number of false positives that they produce [36].

Moreover, the approaches in both trends can be differentiated by observing the way in which the information about road surface conditions is collected. We can distinguish three methods of information collection: mechanical, optical, and vibration-based methods. The mechanical-based methods collect information through mechanical devices that are usually placed in the suspension system [47]. The second group includes approaches that collect information through laser scanners [15, 23, 26] and cameras

[4, 14, 21, 22]. All these approaches require the use of costly and sophisticated instruments; labor-intensive and time-consuming tasks; and professional knowledge [27].

In contrast, the vibration-based methods use accelerometers in order to detect road anomalies. Although some vibration-based methods have proposed the use of dedicated devices to gather the accelerometer information [5, 32, 47], most have adopted a crowdsensing perspective by using smartphones [2, 10, 27, 30, 33, 36, 42, 43, 48, 49]. These approaches take advantage of the massiveness of the smartphones and the sensors they incorporate. In particular, built-in smartphone accelerometers have been utilized to detect potholes and classify the road roughness. These approaches have the advantage of being able to collect information of an entire city daily in little time, due to the fact that many vehicles travel through a city every day, just using the smartphones of the drivers. However, the accelerometers are sensitive to the jitters produced not only by potholes, but also by speed bump and driver actions. For this reason, some works discard some kinds of road surface due to the false-positive results produced. For example, in [33], cobblestone segments from the dataset were discarded; and in [36], the expansion joints present in concrete panels produced a large number of false positives.

Regarding the methods applied to process the accelerometer information, we can distinguish (a) threshold techniques and (b) feature selection and classification methods. The first approaches detect events when the oscillation of the accelerometer exceeds a certain threshold [5, 16, 23]. The second group applied future extraction and selection methods, and then they use classifiers to determine if there are interesting events or not [15, 24, 30, 33, 36, 42]. One limitation that can exhibit these approaches is that collected data can be different depending on the types of car. However, the literature showed that these differences can be settled thanks to the large volume of information that can be obtained using a crowdsensing-based approach [27].

In summary, although many works have addressed the problem of road roughness classification and automatic pothole detection in the last years, some points have not yet been resolved. In particular, within the crowdsensing-based approaches, these points include (a) to automatically detect the kinds of road surface independently of their roughness; and (b) to automatically differentiate stability events produced by man-made structures and driver actions from real potholes. In this sense, our work contributes in both points.

3 Approach

This section is organized as follows. Section 3.1 introduces how the information used by our approach is acquired by using a mobile application. Finally, Section 3.2 describes the proposed approach to road surface classification and pothole detection using deep learning.

3.1 Information acquisition

When a vehicle travels along a road, the kind of road surface, potholes, and other man-made structures affect its stability. These variations in vehicle stability can be detected by the accelerometers of a mobile device (i.e., a smartphone). A built-in smartphone accelerometer provides data on the acceleration of the three coordinate axes with (almost) continuous updates. This allows us to detect the slightest movements. Thus, placing the device in a fixed position inside the vehicle, the mobile movements can reflect the movements of the vehicle. Then, we can detect the presence of potholes or speed bumps in the road, and its severity in terms of destabilization of the vehicle, by analyzing the accelerometer's information. Moreover, if we combine this information with geolocation information, we can determine where these events occur. We name these events as stability events. A stability event is created when there is a difference between the acceleration values sensed by the accelerometer that exceed a given threshold. Thus, when this difference disappears, we assume that the stability event has finished. Since, there are different kinds of road surfaces, it is important that this threshold suits those differences. For this reason, it is important to know the road surfaces by which the accelerometer information was acquired. For example, the threshold for a cobblestone road is clearly different from the threshold for an asphalt road.

The accelerometer information gathered by the mobile application are tuples$acc=\{rawX, diffX, rawY, diffY, rawZ, diffZ\}$. The variables *rawX*, *rawY*, and *rawZ* correspond to the acceleration values sensed by the built-in smartphone accelerometers in the axis X, Y, and Z, respectively. Moreover, the variables *diffX*, *diffY*, and *diffZ* represent the differences between the actual raw values and the previous one. Thus, each travel of a vehicle is a sequence $track = \{(t_1,acc_1), (t_2,acc_2), ..., (t_n,acc_n)\}$ where acc_j is the accelerometer information in time i within the *track*. In particular, we can detect stability events within a track. After losing stability, the vehicle takes several seconds to stabilize again. For this reason, a stability event se is also composed of several tuples acc, such that $se \subset track$.

3.1.1 Road surface classification

To automatically detect the kinds of road surface, we propose analyzing all the accelerometer information stored in a track, without differentiating the existence or absence of a stability event. The kinds of road surface that our approach allow us to classify are concrete panels, cobblestones,

Fig. 1 Road surface classification

asphalt, and dirt roads. Figure 1 shows examples of these four classes. Figure 1a shows a road paved with asphalt, which is the smoothest surface of the four classes. Figure 1b shows a concrete panels road. Although this surface is also smooth, it has expansion joints that impact in the stability of the vehicle. Figure 1c and d show a cobblestone road and a dirt road, respectively. Both of them produce high vibrations in a vehicle, but the cobblestone surface is usually more regular than the dirt one. In this context, one of the hypotheses of our work is that the structural differences of the kinds of road surface produce destabilization patterns that can be classified.

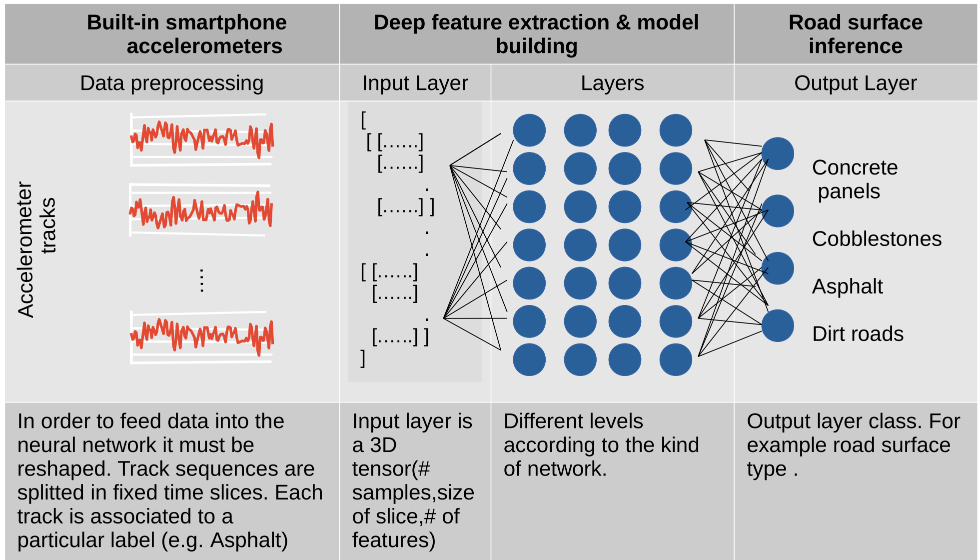
3.1.2 Stability event classification: potholes, street bumps, and driver actions

In contrast to the automatic detection of road surface, to automatically distinguish destabilizations produced by a pothole from destabilizations produced by speed bumps or driver actions, we analyze the stability events along a travel. In particular, we consider two types of speed bumps: speed humps and street gutters (a depression running parallel to a street designed to collect rainwater, but that usually crosses perpendicular streets). Figure 2a and b show two examples of a speed hump and a street gutter, respectively. In addition,

(a) Speed humps (b) Street gutter

Fig. 2 Speed bumps

Fig. 3 Approach



we consider three actions that a driver could perform during a travel, and that they could be wrongly marked as stability events. These driver actions are (a) answering a call; (b) replying to a message; and (c) opening and closing the vehicle's door (when the smartphone is placed in that door).

3.2 Multivariate time series classification with deep learning models

Different techniques have been proposed in the literature for multivariate time series classification using machine learning. However, performance of such techniques mainly depends on good feature extraction methods. Moreover, handcrafting features in a specific application area demands vast domain knowledge and experience and could even hinder their generalization performance [45].

In the last years, the advances in deep learning make it possible to automatically perform high-level feature extraction while achieving promising performance in many areas such as image recognition, machine translation, and speech recognition among others [3, 38, 45]. In particular, CNNs, LSTMs, and RC models have shown improvements over deep neural networks (DNNs) across a wide variety of recognition tasks [37]. Figure 3 shows a conceptual view of our approach for automatic detection of road surface and to automatically distinguish destabilizations produced by potholes. First, at a preprocessing phase, the signal data is segmented in order to feed the network. Moreover, a data augmentation process is also applied to increase the number of samples resulting in a higher model generalization. These topics are discussed in detail in Sections 3.2.1 and 3.2.2, respectively. After that, the different models are defined, and

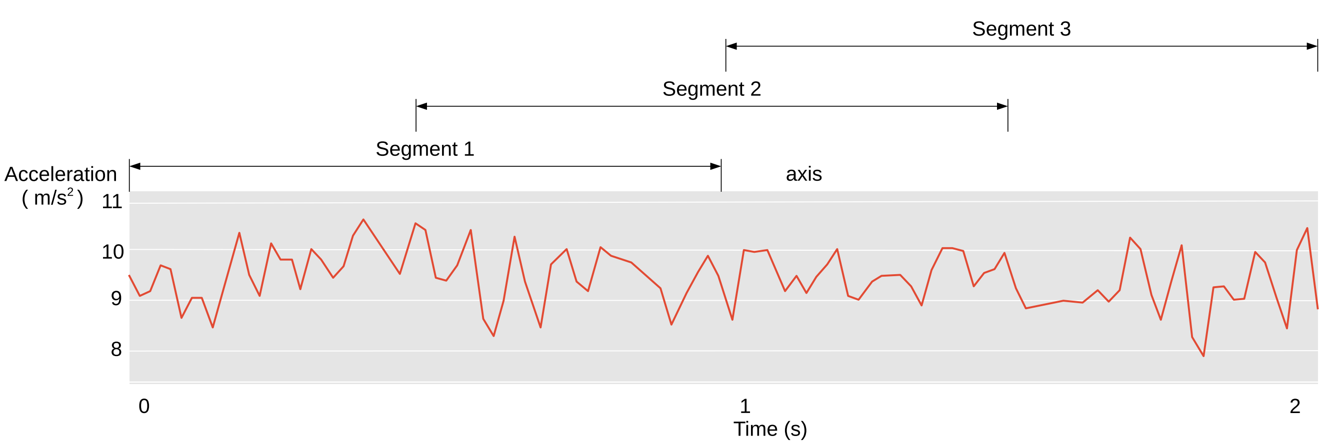


Fig. 4 Fixed-size overlapping sliding window

trained and evaluated applying a cross-validation strategy to estimate the accuracy of the different techniques [29]. In particular, each kind of network, CNN, LSTM, and RC, are described in Sections 3.2.3, 3.2.4, and 3.2.5, respectively.

3.2.1 Data segmentation

First, a data segmentation process is applied to split the sequence of oscillations produced by the accelerometers of the mobile device into several segments as shown in Fig. 4. Then, each segment or features extracted from them are used to train and test the classifiers using supervised machine learning algorithms. Generally, a sliding window is used to segment a data sequence [13]. In particular, we performed an experiment varying the size of the window and selected the best for all techniques, i.e., the size of windows that allows each technique to obtain the best performance described in Section 4.4.1.

3.2.2 Data augmentation

Having few samples to learn from could yield a model that cannot generalize to new data. In order to address this issue, a data augmentation approach can increase the number of samples using random transformations from existing training samples. The objective is to expose the model to more aspects of the data and train a model that generalizes better and avoids overfit [11]. For example, image data augmentation can be performed by shifting the image, zooming, and rotating. In particular, we create new synthetic training samples by stretching or shrinking the sequence of oscillations produced by the accelerometers of the mobile device. In our case, we stretch and shrink training samples altering their frequency (20 ms) randomly between 16 and 24 ms. For example, Fig. 5 shows a z-axis of a "Concrete" sample randomly shrunk.

3.2.3 Convolutional neural network

CNN is a powerful machine learning technique from the field of deep learning. CNNs are an alternative type of neural networks that can be used to reduce spectral variations and model spectral correlations which exist in signals [38]. CNNs have been found highly effective and been commonly used in computer vision and image classification. A CNN is a specialized kind of neural network with a special architecture composed of a sequence of layers of three main types: convolutional, pooling, and fully connected. A convolutional layer captures visual patterns within an image by applying a specified number of filters that scan the input by subregions and perform a set of mathematical operations (convolutions) to generate convolved feature maps. A pooling layer performs a downsampling operation along the spatial dimensions (width, height) of convolutional layers to reduce the dimensionality of feature maps. Finally, a fully connected layer performs classification on the features extracted by the convolutional and pooling layers, connecting every node in this layer to every node in the preceding layer.

In order to apply CNN to acceleration signals, an input adaptation is needed. In particular, the main idea

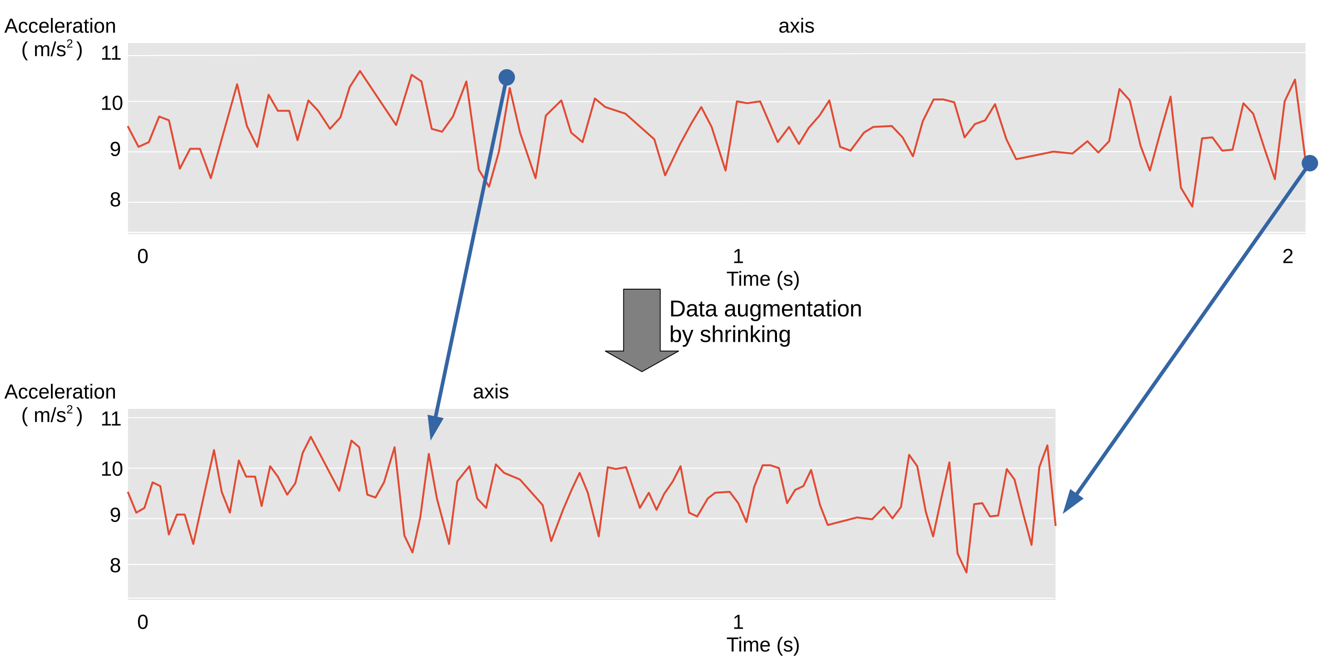
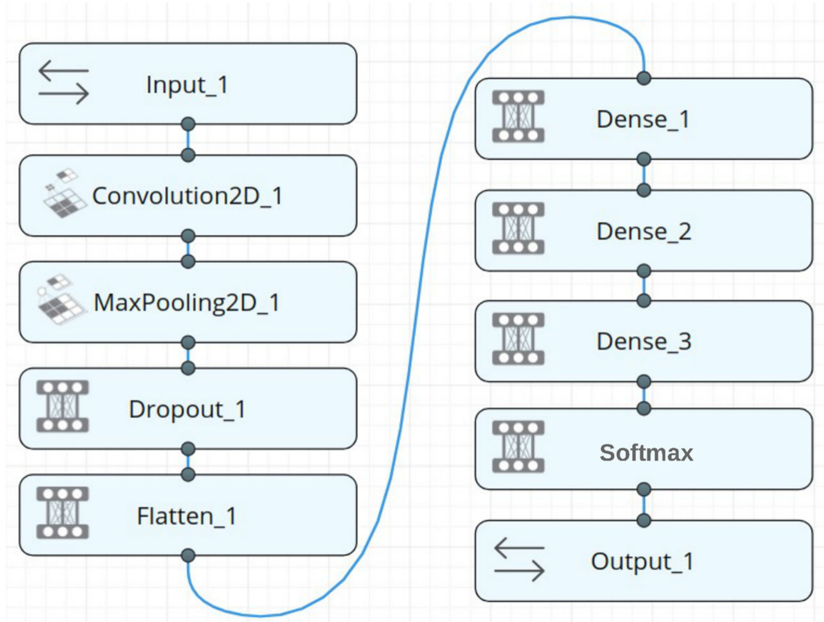


Fig. 5 Data augmentation to enhance model ability to generalize

Fig. 6 CNN model architecture

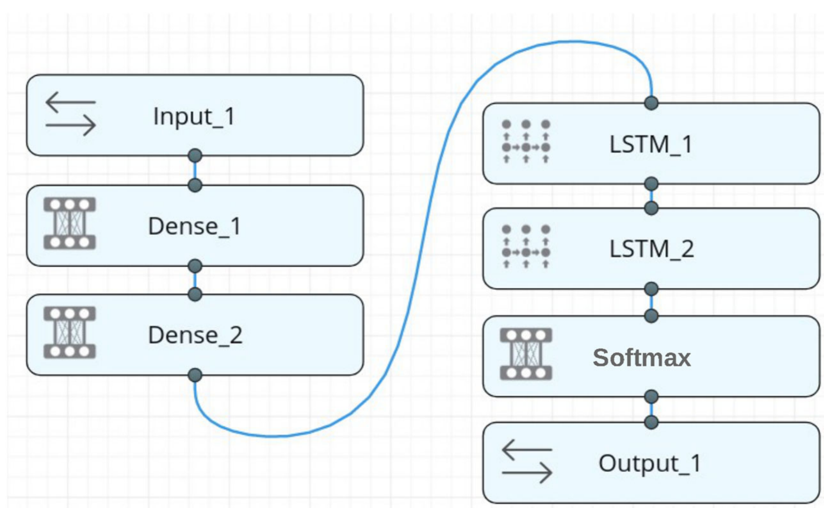


is to transform the inputs (temporal multidimensional 1D samples) to form a *2D virtual image* so as to adapt a 2D convolution [45]. Figure 6 shows the CNN model architecture used in this work. This model has a 2D convolutional layer, followed by a pooling layer, and then a dropout layer for regularization. CNNs learn very quickly, so the dropout layer is intended to help slow down the learning process and hopefully result in a better final model. The pooling layer reduces the learned features in order to consolidate them to only the most essential elements. After that, the learned features are flattened to one long vector and pass through three fully connected layers before the output layer is used to make a prediction. The final layer with a *softmax* function is frequently used as the output of a classifier in order to estimate the probability of each class given a concrete sample.

3.2.4 Long short-term memory network

LSTM is a specific recurrent neural network (RNN) architecture that was designed to model temporal sequences and their long-range dependencies more accurately than conventional RNNs [39]. The LSTM contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to

Fig. 7 LSTM model architecture



special multiplicative units called gates to control the flow of information. LSTM has a special architecture which enables it to forget the unnecessary information. Figure 7 shows the LSTM model architecture used. In particular, this model has 2 fully connected layers, followed by 2 LSTM layers, and then a final layer with a *softmax* function.

3.2.5 Reservoir computing

RC is a family of RNN models. Its recurrent part is always kept fixed and it can be either generated randomly or by using custom topologies to make the information flow easier [28, 34]. Notwithstanding this architectural simplification, the recurrent part of such reservoir delivers a large dynamic feature supply that helps to resolve a great variety of tasks. Besides, RC models can achieve outstanding performances in many fields, including time series forecasting [6, 9, 25], process modeling [34], and speech analysis [44]. As the reservoir is fixed, it is needed to only train the readout, which provides an immediate mapping between the internal representation of the reservoir and the task-specific output.

So, the reservoir computing section has been mainly brought up to make the most of the RNN's computational power, without the need of training previously the internal weights. Instead, from this point of view, the reservoir behaves as a complex non-linear dynamic filter that transforms the input signals just by using a high-dimensional temporal map, not unlike the operation of an explicit, temporal kernel function [41].

The generalization capabilities are based on three components: (i) a large number of processing units in the recurrent layer, (ii) sparsity of the recurrent connections, and (iii) a spectral radius of the connection weights matrix Wr, set to bring the system to the edge of stability [7]. Therefore, the way the reservoir behaves can be controlled by modifying the following structural hyperparameters, instead of training the internal weight matrices: the spectral radius ρ; the percentage of non-zero connections β; and the number of hidden units R.

In this paper, we applied the approach proposed in [8] as shown in Fig. 8. Such approach consists of four modules, which have been instantiated as follows: a bidirectional reservoir configuration; a dimensionality reduction based on principal component analysis (PCA) that applies a reduction on the reservoir's states produced sequence; a reservoir model that delineates the correct way to represent the input time series from the reservoir's state sequence; and a linear model for performing the ultimate classification.

4 Experimental evaluation

4.1 Experimental settings

To evaluate our proposal, the experiments were carried out with real-world information extracted from Tandil, Buenos Aires, Argentina. It is worth noticing that all the information were obtained from routine travels in different road conditions, that is, it was not obtained in a controlled scenario. In total, 97 travels were processed. From these travels, we identify 26.78 km of asphalt roads, 24.12 km of concrete roads, 12.65 km of cobblestone roads, and 16.95 km of dirt roads.

Moreover, we identified 24 potholes, 55 speed humps, and 46 street gutters in the streets through which the vehicle traveled. Then, taking into account the potholes and the speed bumps identified, we extracted 618 stability events, particularly 302 events associated with potholes, 218 events

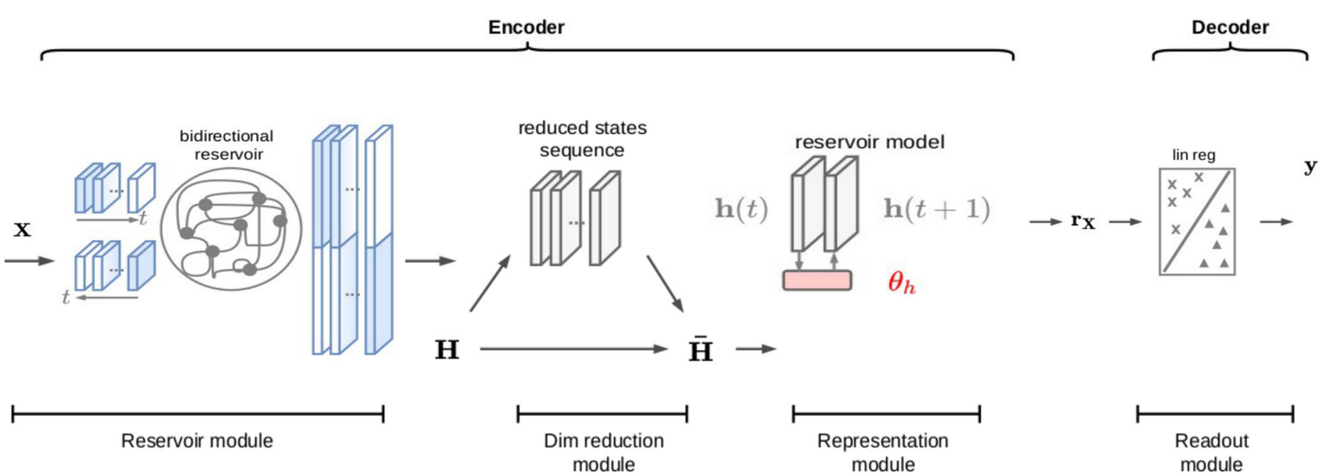
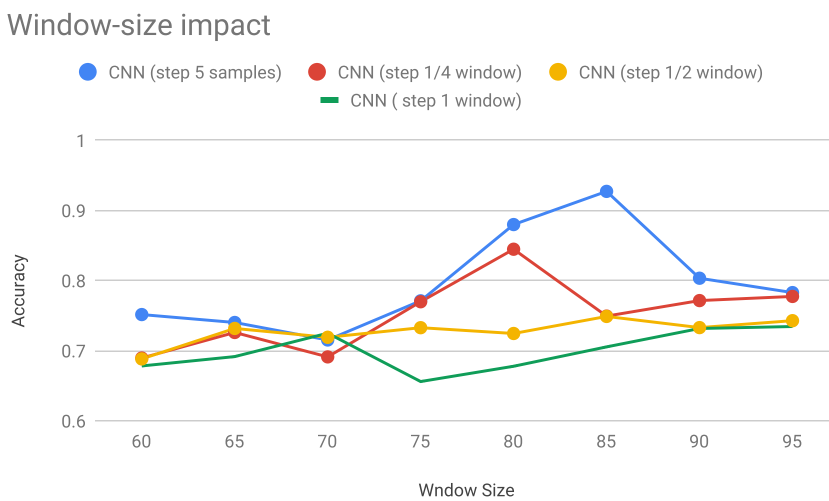


Fig. 8 RC model architecture (adapted from [8])

Fig. 9 Accuracy analysis to determine window size for road roughness classification



associated with speed humps, and 98 events associated with street gutters. Additionally, we identified 29 calls answered, 49 messages replied, and 34 events produced by door opening/closing. In average, each event was compound of 19.68 tuples *acc*.

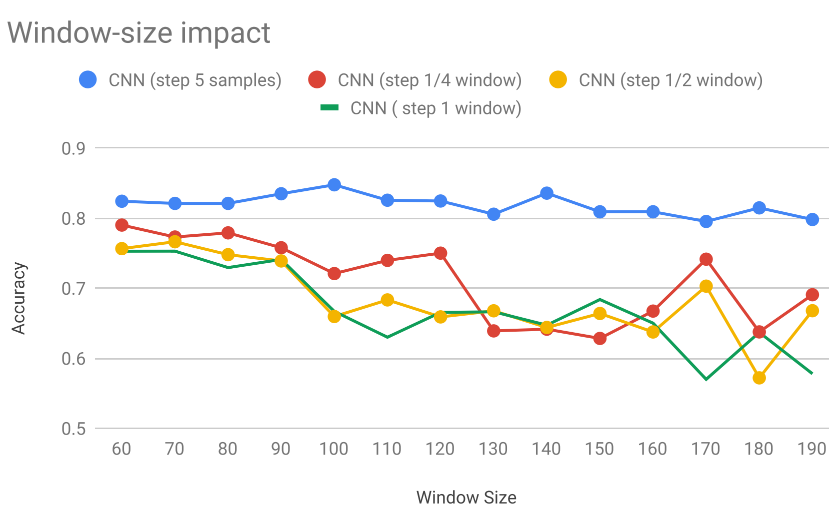
4.2 Procedure

The experimental evaluation was divided into two parts. The first part consisted of classifying the kind of road surface, and the second one consisted of classifying each stability event according to its origin. However, some tasks of preprocessing were carried out on both datasets. As described below, we applied some techniques for data augmentation and segmentation on the original data. The aim of applying these techniques was to increment the number of training and testing data and to keep balanced the classes in the dataset. After applying these techniques, we obtained 2352 stretches of "Asphalt" class, 2319 stretches of "Cobblestone" class, 2181 stretches of "Concrete" class, and 2363 stretches of "Dirt" class.

Regarding the second problem, we obtained 275 events associated with class "Calls," 302 events associated with class "Door," 263 events associated with class "Message," 252 events associated with class "Pothole," 256 events associated with class "Speed Hump," and 260 events associated with class "Street Gutter."

Then, to train and test our approach, we applied a cross-validation technique 10-fold. To do this, we took into account that if two events *a* and *b* were produced from the same original event (i.e., through augmentation or segmentation), *a* and *b* must belong to the same fold. Finally, we compare

Fig. 10 Accuracy analysis to determine window size for automatic pothole detection



the proposed models with other baseline models (see next subsection) taking into account 4 well-known metrics: accuracy, precision, recall, and F-measure.

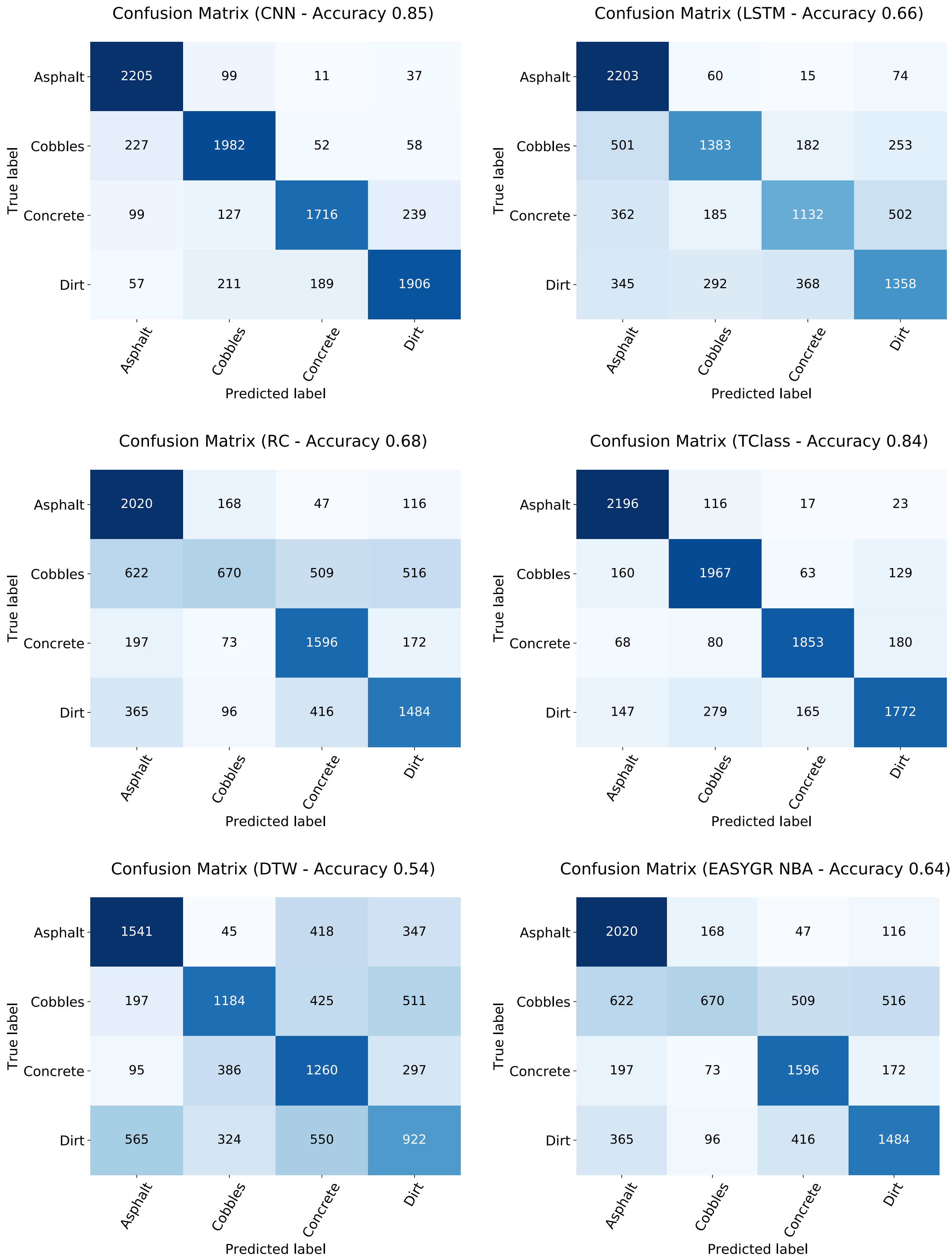
4.3 Baselines

As baselines, we selected three non-neural network techniques:

– TClass (TC) [20] proposes a feature construction technique that parameterizes sub-events of the training set and clusters them to construct features. Once the features are obtained, a standard classifier is built to classify new instances. Some of the components that can be applied to construct features are global extractors (duration, mean, minimum and maximum and mode of a variable of the sequence), and meta-features (increasing, decreasing, plateau, localmax, localmin). Regarding the results reported for TClass, we evaluated different configurations. In consequence, we reported just the best results that were obtained by using K-nearest neighbors classifier (IBk implementation from Weka) and by using the global extractors *duration*, *mean*, *min*, and *max* over the 6 attributes of the sequences: *rawX*, *diffX*, *rawY*, *diffY*, *rawZ*, and *diffZ*.
– Dynamic Time Warping (DTW) finds the similarity between two time series by aligning them optimally [40]. A time series is an ordered sequence of values measured at equally spaced time intervals. In the context of accelerometer analysis, the series are represented by the track sequences that describe each acceleration value in a certain axis. To align two sequence tracks, DTW iteratively stretches and shrinks the time axis so as to find the minimum distance between each pair of points in the tracks. Applying DTW results in a distance value that measures the similarity between the tracks. After applying the algorithm between each pair of tracks in the set, we obtain a model track that represents the set and the acceptance threshold. The model track is the one that resembles the rest of the tracks the most. We select it by adding all the distances from each track to the rest and selecting the track that gets the lowest distance. The upper limit of the acceptance threshold is given by the longest distance between the tracks that differs the most. Therefore, we set the acceptance threshold between 0 and this value.
– EasyGR-NB (EGR) is a gesture recognition technique based on Naïve Bayes and K-means [17]. Naïve Bayes is a simple probabilistic classifier based on the Bayes' theorem with strong (naive) independence assumptions [35]. A classifier is used to find a dependent variable or class based on some observed variables. The problem with this definition is that when the number of

Table 1 Precision, recall, and F-measure for road surface classification

Class	Precision						Recall						F-measure					
	CNN	LSTM	RC	TC	DTW	EGR	CNN	LSTM	RC	TC	DTW	EGR	CNN	LSTM	RC	TC	DTW	EGR
Asphalt	0.85	0.65	0.63	0.85	0.64	0.63	0.94	0.94	0.94	0.93	0.66	0.86	0.89	0.76	0.76	0.89	0.65	0.73
Cobbles	0.82	0.72	0.84	0.81	0.61	0.67	0.85	0.6	0.62	0.85	0.51	0.29	0.84	0.65	0.72	0.83	0.56	0.4
Concrete	0.87	0.67	0.56	0.88	0.47	0.62	0.79	0.52	0.6	0.85	0.62	0.78	0.83	0.58	0.58	0.87	0.54	0.69
Earth	0.85	0.62	0.79	0.84	0.44	0.65	0.81	0.57	0.55	0.75	0.39	0.63	0.83	0.6	0.65	0.79	0.42	0.64
Total	0.85	0.66	0.68	0.84	0.54	0.64	0.85	0.66	0.68	0.84	0.54	0.64	0.85	0.65	0.68	0.84	0.54	0.61



CNN	Asphalt	Cobbles	Concrete	Dirt
Asphalt	2205	99	11	37
Cobbles	227	1982	52	58
Concrete	99	127	1716	239
Dirt	57	211	189	1906

LSTM	Asphalt	Cobbles	Concrete	Dirt
Asphalt	2203	60	15	74
Cobbles	501	1383	182	253
Concrete	362	185	1132	502
Dirt	345	292	368	1358

RC	Asphalt	Cobbles	Concrete	Dirt
Asphalt	2020	168	47	116
Cobbles	622	670	509	516
Concrete	197	73	1596	172
Dirt	365	96	416	1484

TClass	Asphalt	Cobbles	Concrete	Dirt
Asphalt	2196	116	17	23
Cobbles	160	1967	63	129
Concrete	68	80	1853	180
Dirt	147	279	165	1772

DTW	Asphalt	Cobbles	Concrete	Dirt
Asphalt	1541	45	418	347
Cobbles	197	1184	425	511
Concrete	95	386	1260	297
Dirt	565	324	550	922

EASYGR NBA	Asphalt	Cobbles	Concrete	Dirt
Asphalt	2020	168	47	116
Cobbles	622	670	509	516
Concrete	197	73	1596	172
Dirt	365	96	416	1484

Fig. 11 Confusion matrixes for road surface classification

observed variables increases, a model based on probability tables becomes infeasible. Therefore, the Bayes' theorem allows developers to decompose the conditional probability making the problem tractable. In this example, the sequences of relative frequencies are used to compute the variance and the mean assuming a normal distribution, which will be used to recognize new sequence tracks. In particular, to apply this algorithm, the tracks need to be encoded as finite-state sequences. Therefore, the k-means algorithm [19] is applied to group the values of the accelerations of the track sequences that belong to the same region in k clusters.

4.4 Results

In this section, we first report the experiments carried out to determine the best size of windows and overlapping step of the data segmentation technique. Then, we present the results of the road surface and the stability event classification.

4.4.1 Analysis of window and overlapping size for data segmentation

We performed some experiments in order to determine the best parameters of the data segmentation technique. In particular, we compared several window and overlapping sizes. For example, Figs. 9 and 10 show the accuracy while varying the size of windows (X-axis) from 60 to 95, and 60 to 190, respectively. Moreover, we explored two different types of sliding windows: Fixed-size Non-overlapping Sliding Window (step: 1 window) and Fixed-size Overlapping Sliding Window (steps: 1/4 window, 1/2 window and 5 samples) [13]. Finally, we selected a 85 window size (1.9 s) for automatic pothole detection, and a 100 window size (2 s) for road roughness classification with a 5-sample step in both cases. The results reported in the next sections were obtained by using these parameter values.

4.4.2 Road surface classification

Table 1 shows the metrics obtained by each evaluated model for road surface classification. As we can see, the best metrics were obtained by CNN. This model obtained an accuracy, precision, recall, and F-measure of 0.85. CNN generously overcame the other deep learning approaches. Figure 11 shows the confusion matrixes for each evaluated model, where we can also observe the accuracy value.

4.4.3 Stability event classification

Table 2 shows the metrics obtained during the stability event classification. Similar to the previous results, CNN obtained the best results generously overcoming the results obtained

Table 2 Precision, recall, and F-measure for stability event classification

Class	Precision						Recall						F-measure					
	CNN	LSTM	RC	TC	DTW	EGR	CNN	LSTM	RC	TC	DTW	NBA	CNN	LSTM	RC	TC	DTW	EGR
Call	0.96	0.76	1	0.97	0.87	0.89	0.93	0.89	0.91	0.85	0.81	0.95	0.94	0.82	0.95	0.9	0.84	0.92
Door	0.94	0.94	0.93	0.96	0.92	0.93	1	0.99	1	0.87	0.99	0.97	0.97	0.96	0.96	0.91	0.96	0.95
Message	0.92	0.95	0.91	0.78	0.79	0.92	0.94	0.9	0.95	0.92	0.93	0.89	0.93	0.92	0.93	0.85	0.86	0.9
Pothole	0.88	0.83	0.87	0.84	0.89	0.86	0.98	1	0.93	0.94	0.99	0.9	0.93	0.91	0.9	0.89	0.94	0.88
S. Hump	0.90	0.6	0.38	0.51	0.53	0.5	0.78	0.33	0.64	0.67	0.68	0.8	0.84	0.43	0.48	0.58	0.6	0.62
S. Gutter	0.95	0.73	0.08	0.68	0.63	0.76	0.93	0.78	0.02	0.44	0.28	0.22	0.94	0.76	0.03	0.53	0.39	0.35
Total	0.93	0.81	0.71	0.8	0.78	0.82	0.93	0.82	0.75	0.78	0.78	0.79	0.93	0.8	0.72	0.78	0.77	0.78

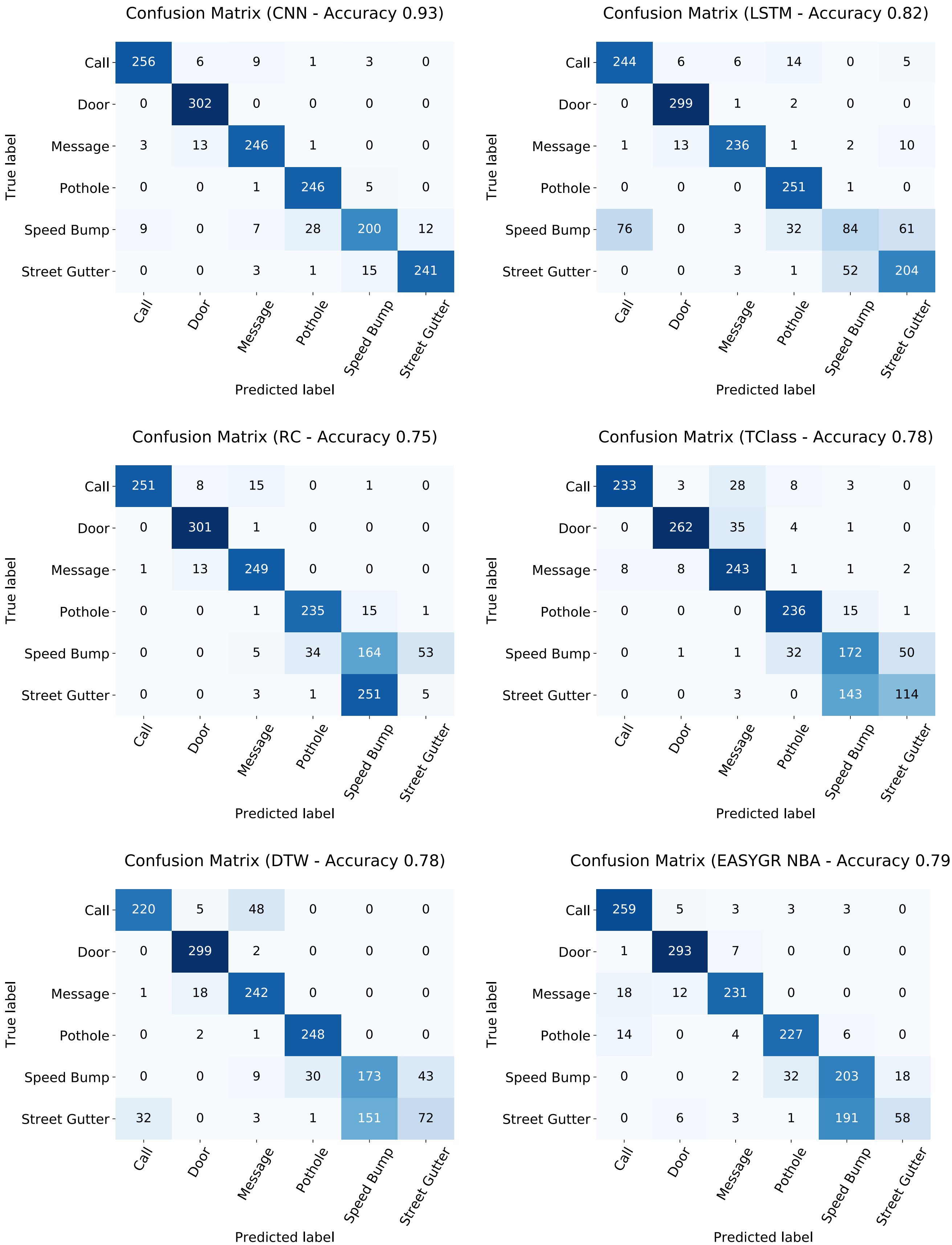


Fig. 12 Confusion matrixes for stability event classification

by the other models. These results clearly showed that it was possible to distinguish potholes from man-made structures and driver actions. Figure 12 shows the confusion matrices and accuracy for each evaluated model. In that figure, we can observe that most potholes were correctly classified, and just a small portion of speed humps were misclassified as potholes using CNN model. We think that this small error could be solved comparing the results of several events gathered in the same geolocated point. Regarding the recall in class Pothole, all the models obtained high values. However, in contrast to CNN, the other models failed to classify the rest of the classes, especially Speed Hump and Street Gutter. This fact is clearly showed by the precision and recall in class Street Gutter obtained by the RC model.

In Sections 3.2.1 and 3.2.2, we also showed the importance of data augmentation and segmentation technique regarding the accuracy of the deep learning models. However, these techniques also had a positive effect in the rest of the models. For example, in [31], we present preliminary results using TClass, but without using data augmentation and segmentation technique. In that work, TClass obtained a precision of 0.68 in class Pothole. In contrast, applying these techniques, TClass obtained a precision of 0.84 in such a class.

Moreover, it is worth noticing that the main aim of this approach is to distinguish potholes from no-potholes events. Taking into account this goal, it is interesting to analyze the accuracy of CNN from this perspective. Thus, considering just two classes (Pothole and No-pothole), the accuracy of CNN was 0.98. Figure 13 shows the confusion matrix and accuracy for Pothole/No-pothole classification.

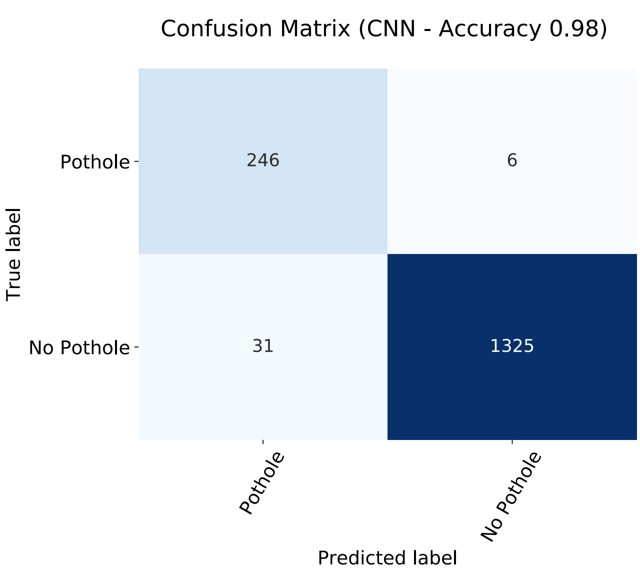


Fig. 13 Confusion matrix and accuracy for pothole/no-pothole classification

5 Conclusions and future work

In this work, we have proposed an approach to automatically identify certain situations that produce a large number of false-positive results in the crowdsensing-based approaches for pothole detection and road surface classification. First, we have proposed an approach to classify the roads according to their surface. This is useful to automatically adjust the thresholds usually adopted to discard noise and detect significant events. This is also important due to the fact that it is hard to find public information about road surfaces. Second, we have proposed an approach to distinguish stability events produced by potholes from stability events produced by other man-made structures or driver actions. This approach is key to improve the accuracy of pothole detection. In fact, the experiments showed a high accuracy in real-world scenarios extracted from routine travels. Thus, it is important to remark due to the fact that most of the experiments in the related works were conducted in controlled scenarios. Moreover, we have shown the importance of data segmentation and augmentation techniques in order to diversify the available information and keep the dataset balanced. Finally, it is worth noticing that both analyzed problems are key if we want to make the application available in multiple cities with the least effort.

Future work will focus on analyzing other kinds of road surfaces and also studying how to use this information to adjust automatically the stability event detection. Other future works will include enriching the pothole detection process by using user profiles. This will allow us to capture differences in the way users drive that could affect the stability of the vehicle.

Acknowledgments We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

1. ASTM E867 - 06 (2012) standard terminology relating to Vehicle-Pavement systems. https://www.astm.org/Standards/E867.htm. Accessed: 2018-10-23
2. Alessandroni G, Carini A, Lattanzi E, Freschi V, Bogliolo A (2017) A study on the influence of speed on road roughness sensing The SmartRoadSense case. Sensors 17(2):1–20
3. Alvarez JR, Arroqui M, Mangudo P, Toloza J, Jatip D, Rodriguez JM, Teyseyre A, Sanz C, Zunino A, Machado C, Mateos C (2018) Body condition estimation on cows from depth images using convolutional neural networks. Comput Electron Agric 155:12–22
4. Beckman GH, Polyzois D, Cha Y-J (2019) Deep learning-based automatic volumetric damage quantification using depth camera. Autom Constr 99:114–124

5. Bello-Salau H, Aibinu AM, Onumanyi AJ, Onwuka EN, Dukiya JJ, Ohize H (2018) New road anomaly detection and characterization algorithm for autonomous vehicles. Applied Computing and Informatics
6. Bianchi FM, De Santis E, Rizzi A, Sadeghian A (2015) Short-term electric load forecasting using echo state networks and pca decomposition. IEEE Access 3:1931–1943
7. Bianchi FM, Livi L, Alippi C (2018) Investigating echo-state networks dynamics by means of recurrence analysis. IEEE Trans Neural Netw Learn Syst 29(2):427–439
8. Bianchi FM, Scardapane S, Løkse S, Jenssen R (2018) Reservoir computing approaches for representation and classification of multivariate time series. arXiv:1803.07870
9. Bianchi FM, Scardapane S, Uncini A, Rizzi A, Sadeghian A (2015) Prediction of telephone calls load using echo state network with exogenous variables. Neural Netw 71:204–213
10. Carrera F, Guerin, S, Thorp JB (2013) By the people, for the people: the crowdsourcing of "Streetbump": an automatic pothole mapping app. ISPRS - Int Arch Photogramm Remote Sens Spat Inf Sci XL-4/W1:19–23
11. Chollet F (2017) Deep Learning with Python, 1st edn. Manning Publications Co., Greenwich
12. Douangphachanh V, Oneyama H (2013) Estimation of road roughness condition from smartphones under realistic settings. In: 2013 13th International Conference on ITS Telecommunications (ITST)
13. Edy SP, Vesilo R (2017) Window-size impact on detection rate of wearable-sensor-based fall detection using supervised machine learning. In: 2017 IEEE life sciences conference (LSC). IEEE, pp 21–26
14. Ghanta S, Shamsabadi SS, Dy J, Wang M, Birken R (2015) A hessian-based methodology for automatic surface crack detection and classification from pavement images. In: Structural health monitoring and inspection of advanced materials, Aerospace, and Civil Infrastructure 2015
15. Gorges C, Öztürk K, Liebich R (2019) Impact detection using a machine learning approach and experimental road roughness classification. Mech Syst Signal Process 117:738–756
16. Harikrishnan H, Gopi VP (2017) Vehicle vibration signal processing for road surface monitoring. IEEE Sens. J. 17(16):5192–5197
17. Ibañez R, Soria A, Teyseyre AR, Berdun L, Campo M (2016) A comparative study of machine learning techniques for gesture recognition using kinect. In: Handbook of Research on Human-Computer Interfaces, Developments, and Applications. IGI Global, pp 1–22
18. Ibańez R, Soria Á, Teyseyre AR, Campo M (2014) Easy gesture recognition for kinect. Adv Eng Softw 76:171–180
19. Jain AK (2010) Award winning papers from the 19th International Conference on Pattern Recognition (ICPR). Pattern Recogn Lett 31(8):651 – 666
20. Kadous MW, Sammut C (2005) Classification of multivariate time series and structured data using constructive induction. Mach Learn 58(2-3):179–216
21. Kertesz I, Lovas T, Barsi A (2007) Measurement of road roughness by low-cost photogrammetric system, pp 36
22. Koch C, Brilakis I (2011) Pothole detection in asphalt pavement images. Adv Eng Inform 25(3):507–515
23. Kumar P, Angelats E (2017) An automated road roughness detection from mobile laser scanning data. ISPRS - International Archives of the Photogrammetry. Remote Sens Spatial Inf Sci XLII-1/W1:91–96
24. Lepine J, Rouillard V, Sek M (2018) Evaluation of machine learning algorithms for detection of road induced shocks buried in vehicle vibration signals. Proc. Inst. Mech. Eng. Pt. D J. Automobile Eng., pp 095440701875620
25. Li D, Han M, Wang J (2012) Chaotic time series prediction based on a novel robust echo state network. IEEE Trans Neural Netw Learn Syst 23(5):787–799
26. Li Q, Yao M, Yao X, Xu B (2009) A real-time 3D scanning system for pavement distortion inspection. Measur Sci Technol 21(1):015702
27. Li X, Goldberg DW (2018) Toward a mobile crowdsensing system for road surface assessment. Comput Environ Urban Syst 69:51–62
28. Lukoševičius M, Jaeger H (2009) Reservoir computing approaches to recurrent neural network training. Comput Sci Rev 3(3):127–149
29. Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine Learning, Neural and statistical classification. Ellis Horwood
30. Mohamed A, Fouad MMM, Elhariri E, El-Bendary N, Zawbaa HM, Tahoun M, Hassanien AE (2015) RoadMonitor: An intelligent road surface condition monitoring system. In: Advances in Intelligent Systems and Computing, pp 377–387
31. Monteserin A (2018) Potholes vs. speed bumps: A multivariate time series classification approach. In: Proceedings of the 1st Workshop on Knowledge Discovery and User Modelling for Smart Cities co-located with 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, UMCit@KDD 2018, London, pp 36–40
32. Ngwangwa HM, Heyns PS (2014) Application of an ANN-based methodology for road surface condition identification on mining vehicles and roads. J Terramech 53:59–74
33. Perttunen M, Mazhelis O, Cong F, Kauppila M, Leppänen T, Kantola J, Collin J, Pirttikangas S, Haverinen J, Ristaniemi T, Riekki J (2011) Distributed road surface condition monitoring using mobile phones. In: Lecture Notes in Computer Science, pp 64–78
34. Rodan A, Sheta AF, Faris H (2017) Bidirectional reservoir networks trained using svm+ privileged information for manufacturing process modeling. Soft Comput 21(22):6811–6824
35. Russell S, Norvig P (2009) Artificial Intelligence: A Modern Approach, 3rd edn. Prentice Hall Press, Upper Saddle River
36. Md S, Mandal S, Islam S (2015) A support vector method for automated road anomaly detection using mobile device. Int J Comput Appl Technol 127(9):16–19
37. Sainath T, Vinyals O, Senior A, Sak H (2015) Convolutional, long short-term memory, fully connected deep neural networks. In: ICASSP
38. Sainath TN, Kingsbury B, Saon G, Soltau H, Mohamed A-r, Dahl G, Ramabhadran B (2015) Deep convolutional neural networks for large-scale speech tasks. Neural Netw 64(C):39–48
39. Sak H, Senior AW, Beaufays F (2014) Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv:1402.1128
40. Salvador S, Chan P (2007) Toward accurate dynamic time warping in linear time and space. Intell Data Anal 11(5):561–580
41. Schrauwen B, Verstraeten D, Van Campenhout J (2007) An overview of reservoir computing: theory, applications and implementations. In: Proceedings of the 15th European Symposium on Artificial Neural Networks. pp 471–482
42. Silva N, Shah V, Soares J, Rodrigues H (2018) Road anomalies detection system evaluation. Sensors 18(7):1–20
43. Tai Y-C, Chan C-W, Hsu JY-J (2010) Automatic road anomaly detection using smart mobile device. In: 2010 15th Conference on Artificial Intelligence and Applications (TAAI)
44. Trentin E, Scherer S, Schwenker F (2015) Emotion recognition from speech signals via a probabilistic echo-state network. Pattern Recogn Lett 66:4–12
45. Wang J, Chen Y, Hao S, Peng X, Hu L (2017) Deep learning for sensor-based activity recognition: A survey. arXiv:1707.03502

46. Xing Z, Pei J, Keogh E (2010) A brief survey on sequence classification. SIGKDD Explor Newsl 12(1):40–48
47. Yaqub R, Burrell J, Crutcher X, Morrow J (2017) Prototype model of potholes and road-roughness detection and reporting system. In: 2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)
48. Zeng H, Park H, Smith BL, Parkany E (2017) Feasibility assessment of a Smartphone-Based application to estimate road roughness. KSCE J Civ Eng 22(8):3120–3129
49. Zhang Z, Deng F, Huang Y, Bridgelall R (2015) Road roughness evaluation using in-pavement strain sensors. Smart Mater Struct 24(11):115029
50. Zhang Z, Sun C, Bridgelall R, Sun M (2018) Application of a machine learning method to evaluate road roughness from connected vehicles. J Transp Eng Part B: Pavements 144(4):04018043