applied sciences

MDPI

Communication

On Training Road Surface Classifiers by Data Augmentation

Addisson Salazar [1], Alberto Rodríguez [2], Nancy Vargas [1] and Luis Vergara [1,*]

[1] Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, C/Camino de Vera s/n, 46022 València, Spain; asalazar@dcom.upv.es (A.S.); nanvarme@doctor.upv.es (N.V.)
[2] Departamento de Ingeniería de Comunicación, Universidad Miguel Hernández, Avda. de la Universitat d'Elx s/n, 03202 Alicante, Spain; arodriguezm@umh.es
* Correspondence: lvergara@dcom.upv.es

Abstract: It is demonstrated that data augmentation is a promising approach to reduce the size of the captured dataset required for training automatic road surface classifiers. The context is on-board systems for autonomous or semi-autonomous driving assistance: automatic power-assisted steering. Evidence is obtained by extensive experiments involving multiple captures from a 10-channel multi-sensor deployment: three channels from the accelerometer (acceleration in the X, Y, and Z axes); three microphone channels; two speed channels; and the torque and position of the handwheel. These captures were made under different settings: three worm-gear interface configurations; hands on or off the wheel; vehicle speed (constant speed of 10, 15, 20, 30 km/h, or accelerating from 0 to 30 km/h); and road surface (smooth flat asphalt, stripes, or cobblestones). It has been demonstrated in the experiments that data augmentation allows a reduction by an approximate factor of 1.5 in the size of the captured training dataset.

Keywords: driving assistance; road surface classification; machine learning; data augmentation

check for updates

Citation: Salazar, A.; Rodríguez, A.; Vargas, N.; Vergara, L. On Training Road Surface Classifiers by Data Augmentation. *Appl. Sci.* **2022**, *12*, 3423. https://doi.org/10.3390/app12073423

Academic Editor: Zhijun Chen

Received: 25 February 2022
Accepted: 25 March 2022
Published: 28 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Preliminaries

Road surface automatic analysis has a long history of developments involving a variety of signal modalities and their corresponding sensors. Most previous works have been centered on estimating road roughness into the levels defined in ISO 8606/1995 or detecting anomalies in the road (potholes, bumps, cracks, etc.) for supervising and planning road maintenance. References [1–5] are some representative examples. However, during the last few years, the emergence of autonomous or semi-autonomous car technology has strongly increased the interest in on-board systems, which can perform real-time surface classification to aid the driving assistance systems of the car by automatic power-assisted steering. A variety of works have recently been reported in this very active research area [6–17]. Most of them propose machine learning approaches, where an automatic classifier is to be trained from labelled feature vectors derived from sensor signals. Several signal modalities, features, and classifiers have been proposed so far, but there is still a long way to reach efficient solutions demonstrating compatibility with commercial constraints of vehicle mass production.

In the general area of automatic classification, the training set size has been demonstrated to have a large impact on the classifier performance [18–21]; however, in many application domains, practical constraints prevent an arbitrarily large recording of signals. This is the case in road surface classification. Automatic road surface classifiers must be trained off-line before they can be incorporated into the car. Robust operation of the road surface classifier implies that training must take into account different possible conditions: worm-gear interface configuration, hands on or off the wheel, vehicle speed, kind of vehicle, and road surface classes. Moreover, this training could require updating after some period to account for some corrections or to extend the capabilities of the classifier. However, training is a costly process requiring prepared cars and dedicated closed circuits. Signals recorded during the driving of the car over different types of known surfaces can be labelled to obtain the required dataset for training and testing.

One possible approach to alleviate the costly capture of real data is data augmentation. Different methods of general applicability have been proposed in the literature, which use synthetic data to augment the real dataset in an effort to achieve an effective increase in the training set size. Two representative reviews in this regard are references [22,23] and examples of quite different applications are: patent analysis [24], speech processing [25], and medical image analysis [26]. There are a few works devoted to road surfaces where data augmentation has been studied: detection of road surface anomalies (bumps, potholes, etc.) based on the sensor data collected with smartphones [27,28] and image-based road surface classification [29]. Both works are constrained to the imbalance scenario, where only one class is deficient while the others are abundant.

1.2. Novelties and Contribution

The main contribution of this work is to demonstrate the usefulness of data augmentation in road surface classification. Novelties with respect to the few existing related works, as mentioned above, are that we focus on automatic power-assisted steering, that neither smartphones nor cameras are used, and that all classes are assumed deficient. Moreover, a recent method for data augmentation is considered: Generative Adversarial Network Synthesis for Oversampling (GANSO) [30]. This method has demonstrated its superiority regarding classical oversampling methods. It requires the incorporation of structural information of the feature vector to obtain more realistic replicas. This is very specific for every application. Thus, we have achieved the adaptation of GANSO to the problem of data augmentation for road surface classification. We will show that GANSO provides, in all cases, better results than the benchmark method, the Synthetic Minority Oversampling Technique (SMOTE) [22,23].

Therefore, to the best of the authors' knowledge, this communication provides, for the first time, a demonstration of the usefulness of the approach. The results should motivate future research in both theoretical and experimental aspects of this challenging problem.

2. The Road Surface Classification System

Figure 1 shows the main components of the automatic classifier. First, a given number of signals (channels) are collected from different kinds of sensors. Ten channels were considered in our deployment. The first three are the X, Y, and Z channels of an accelerometer on the intermediate shaft. The next three correspond to two microphones located close to the driver's head plus one on the upper side of the electric power steering (EPS) system column. Microphones are obvious candidates to incorporate in the multisensory system for the automatic classification of the type of road surface, as they are the best emulators of the human ear. Two more channels record the speed of the left and right wheels; the last two channels correspond to the measurement of the torque and the position of the handwheel, respectively. Figure 1 also shows a moving time window implemented for every channel. Moreover, a set of features (only four features are shown) are sequentially extracted from the time interval spanned by the moving window. All the features from all the channels corresponding to the same location of the moving window are the elements of a feature vector, as indicated in Figure 1. Dimensionality reduction of the feature vector is recommended to avoid redundancies, noise filtering, and to alleviate the computational burden. The sequences of feature vectors are the input to an automatic classifier, which sequentially selects a road surface class among a given set of options. Thus, road surface profile estimates are continuously given to the driving assistance systems in order to make appropriate adjustments.

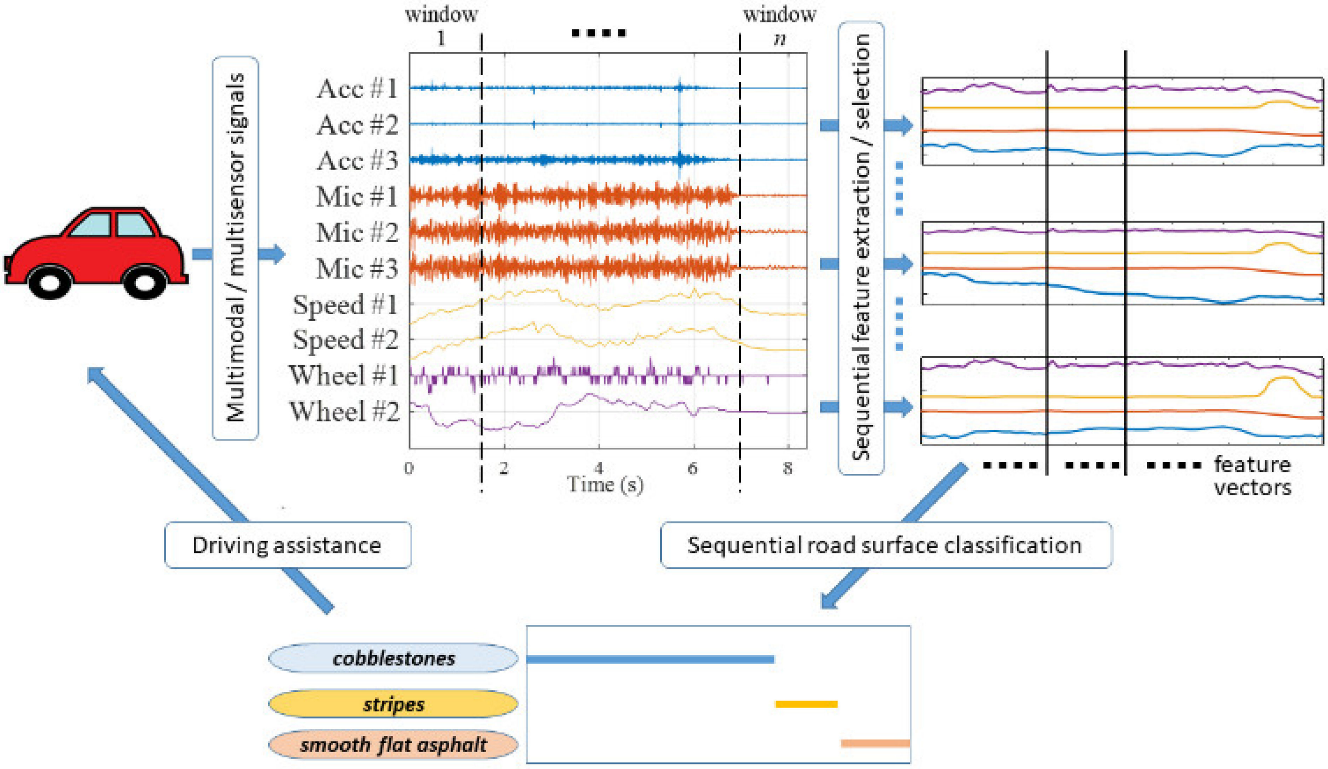

Figure 1. Multisensor road surface classification system.

3. Data Augmentation

A key aspect of the previous scheme is the training of the classifier. In the experiments of Section 4, it is demonstrated that significant saving of signal acquisitions can be obtained by data augmentation. This latter has been implemented by generating synthetic feature vectors that preserve the structural properties of the original features while providing an oversampling of the feature space.

We have considered the method Generative Adversarial Network Synthesis for Oversampling (GANSO) as described in [30]. This method has demonstrated its superiority regarding classical oversampling methods, particularly for small training sets (small data problem), and where all classes could be deficient. Other data augmentation methods are mostly applied for the imbalance problem, where only one class is deficient. Let us comment on the main ideas of GANSO; specific mathematical details of the general method may be found in [30]. Specific details of the adaptation of GANSO to the application considered in this work are given in Section 4. A Generative Adversarial Network is composed by two blocks; the first one (generative) provides examples to the second one (discriminative), which is to decide if the example provided by the generative block should be accepted or rejected. Both blocks collaborate in the sense that the discriminative block "informs about why" the example is rejected so that the generative block can modify the example until it is accepted. This is shown in Figure 2.

One original feature vector (OFV) is provided to the generative block to generate a synthetic feature vector (SFV) by transforming the OFV using the complex graph Fourier transform [31], a mathematical operator developed in the framework of graph signal processing [32]. This requires a graph model of the connections between the different components of the feature vector. Moreover, from this graph model, we compute the maximal cliques (subsets of fully connected vertices that cannot be extended by adding more adjacent vertices). Thus, a structured representation is built into the feature vector space. Then, the maximal cliques of the OFV and SFV are, respectively, correlated with the maximal cliques of a reference subset formed by other OFVs—this is termed structured correlation in Figure 2. By doing so, we obtain two sets of discriminative features, respectively corresponding to the true class and to the candidate class. Then, these two sets are used to train a two-class linear discriminant in the discriminative block. If the two classes cannot be properly separated, SFV is accepted; if not, SFV is rejected. In this latter case, the discriminative block informs the generative block about the linear separator used to discriminate both classes, so that a corrected SFV is provided, trying to obtain a better candidate. All the process is iterated until SFV is accepted.

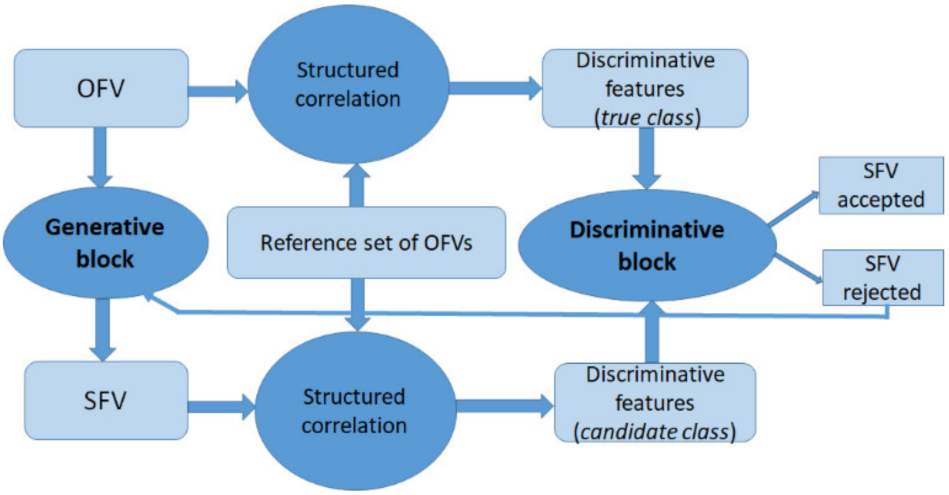

Figure 2. Description of the oversampling method GANSO. OFV, Original Feature Vector; SFV, Synthetic Feature Vector.

4. Experiments

First, let us indicate the main elements of the experimental setup. All 10 channels (described in the previous section) were sampled at 48 KHz. The moving time window duration was selected to be 1.5 s, and the shift between two consecutive windows was 0.1 s. These values were selected in a rather experimental manner to maximize the classification accuracy. In every location of the time window, 56 features were extracted for every channel. Given that most of the channels are audio channels or vibration channels, our tests considered several spectral features that are commonly used for audio classification [33], namely:

- Average power across all frequency bands. We compute 1 feature per channel.
- Spectral contrast: ratio between the minimum and maximum spectral values in each octave. As in most applications on audio processing, the octaves are referenced to 440 Hz. Given the sampling frequency, we had 10 octaves. Therefore, we computed 10 features per channel.
- Spectral slope: first-order polynomial trend of the power spectrum, assuming that the spectrum follows a power law of the frequency. We compute 1 feature per channel.
- Spectral flatness: measure of the variability of the spectrum in a given band, obtained as the ratio of the geometric and arithmetic means of the power spectrum. As per the MPEG7 standard, we considered bands of one quarter of an octave. We had 10 octaves; therefore, we computed 40 features per channel.
- Centroid frequency, 1 feature per channel.
- Maximum frequency, 1 feature per channel.

We also considered the following high-order statistics:

- Third-order autocorrelation, 1 feature per channel.
- Time reversibility, 1 feature per channel.

This resulted in a total of 56 features extracted from each of the 10 available channels, which provides a total of 560 features at every window location. Table 1 shows the mathematical definitions of all features.

Table 1. Extracted feature definitions.

Feature	Definition (In All Cases Δ Is the Number of Samples within the Time Window)
Average power	$\frac{1}{\Delta}\sum_{n=1}^{\Delta} x^2(n)$ **1 feature**
Spectral contrast	$\frac{\max_f \lvert X(f)\rvert}{\min_f \lvert X(f)\rvert}, f_{o1} \leq f \leq f_{o2}$ where f_{o1}, f_{o2} are the start and end indices of oth octaves, respectively. 440Hz was the reference (end limit of the 4th octave) [26], 10 octaves were considered. **10 features**
Spectral slope	Trend a of the model $\log\lvert X(f)\rvert = a\log f + b$, **1 feature**
Spectral flatness	$\frac{\left(\prod_{f=f_{o1}}^{f_{o2}} \lvert X(f)\rvert\right)^{1/(f_{o2}-f_{o1})}}{\frac{1}{f_{o2}-f_{o1}}\sum_{f=f_{o1}}^{f_{o2}} \lvert X(f)\rvert}$ where f_{o1}, f_{o2} are respectively the start and end indices of the oth quarter of octave [26]. 10 octaves were considered. **40 features**
Centroid frequency	$\frac{f_S}{\Delta}\frac{\sum_{f=1}^{\Delta} f\lvert X(f)\rvert^2}{\sum_{f=1}^{\Delta}\lvert X(f)\rvert^2}$ where $X(f)$ is the direct Fourier transform of x within the epoch taken at Δ points, and f_s is the sampling rate. **1 feature**
Maximum frequency	$\frac{f_S}{\Delta}\left(\operatorname{argmax}_f \lvert X(f)\rvert\right)$ **1 feature**
Third-order autocorrelation	$\frac{1}{\Delta-2}\sum_{n=3}^{\Delta} x(n)\cdot x(n-1)\cdot x(n-2)$ **1 feature**
Time reversibility	$\left(\frac{1}{\Delta}\sum_{n=1}^{\Delta} x^2(n)\right)^{-3/2} \frac{1}{\Delta-1}\sum_{n=2}^{\Delta}(x(n)-x(n-1))^3$ **1 feature**

The feature ranking method proposed in [13] was applied, retaining the best ranked feature of every channel to get 10 selected features.

Resuming, 56 features are extracted for every sensor signal interval corresponding to the location of the moving window, as described in Figure 1. Then, a feature vector having 56 × 10 = 560 components is computed for every location of the moving window by aggregating the 56 features of the 10 channels. After feature ranking, we only retain the best ranked feature of every channel. Thus, a feature vector of just 10 components is finally computed for every location of the moving window of Figure 1. Thus, we guarantee that all channels are involved in the overall procedure.

Among the large number of possible classifiers, we have considered two methods. The first one is the well-known Linear Discriminant Analysis (LDA) [34]. This is a practical option from the point of view of maximum simplicity for the possible commercial implementation of the on-board road surface classifier. It implies simple linear computations on the elements of the extracted feature vector. Training of LDA is also simple as it amounts to estimating correlations on the training set and solving linear systems of equations. On the other extreme, we have also tested the Random Decision Forest (RDF) classifier [35]. This is a complex method that combines the outputs of many decision trees. Training of the RDF is rather complex as an iterative search is required to obtain optimum decision trees to be combined. Implementation is also complex as every element of the extracted feature vector is to be compared with a properly selected threshold to define every branch of the many decision trees.

A total number of 63 10-channel captures (multichannel signal acquisitions) were made. Each capture corresponded to a straight line path of the car with an average time duration of 14.59 s. The captures considered different configurations of the following four properties: (i) worm-gear interface configuration (three configurations to simulate different noise and vibration conditions on the driver); (ii) hands on or off the wheel; (iii) vehicle speed (constant speed of 10, 15, 20, 30 km/h, or accelerating from 0 to 30 km/h); (iv) road surface (smooth flat asphalt, stripes, or cobblestones).

A dataset consisting of 8039 10-dimensional feature vectors was obtained from the 63 10-channel captures using the setup described above. Notice that the overall duration of the captures is directly related to the size of the dataset. Thus, it will be relevant any reduction in the dataset size, not affecting the training quality.

We define the training set (*TS*) as the set of captured feature vectors from the total set that will be used for training. The cardinality of the *TS* will be denominated training set size (*TSS*). Thus, a variable TSS was reserved for training. A fixed size of 500 was considered for the testing set, not including members of the *TS* to avoid overfitting. Let us now consider the most significant results of the experiment. Figure 3 shows the mean accuracy (percentage of correct classification in the testing set) for TSS varying from 377 to 1505 in steps of 125.

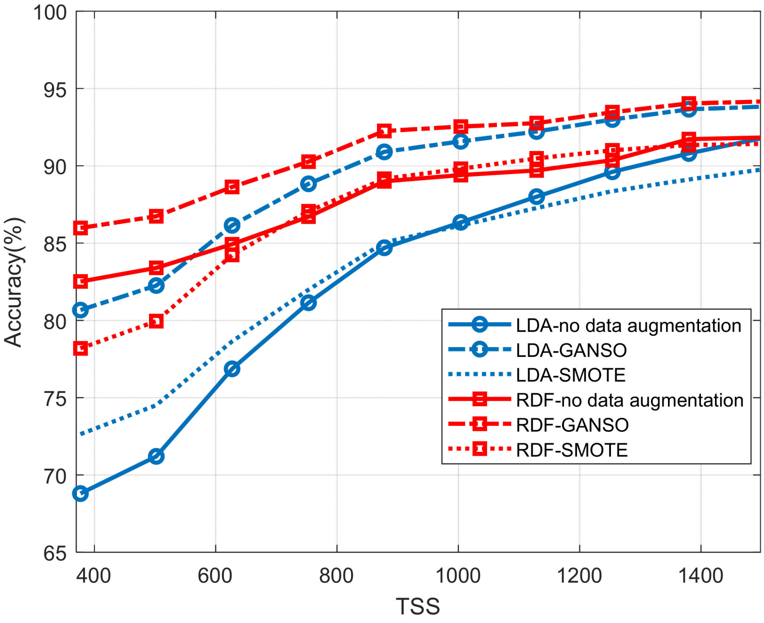

Figure 3. Variation of the accuracy for different values of the training set size (*TSS*).

Mean accuracy was obtained by averaging 100 iterations, where every iteration corresponded to a different random partition of the training and testing sets. In every iteration, accuracy has been computed from sample estimates: quotient between the number of correct road surface classifications and the total number of cases to be classified. Then, it is multiplied by 100 to express it as a percentage.

Figure 3 represents six curves. Three of them (blue) correspond to LDA and the other three (red) to RDF. Notice that for every type of classifier, the solid line corresponds to training without data augmentation, while the dot-dash line and dot line correspond to training with data augmentation, where the TS was duplicated by adding the appropriate number of synthetic 10-dimensional feature vectors. In particular, the dot line curve corresponds to data augmentation using the Synthetic Minority Oversampling Technique (SMOTE) [22,23], which is the most consolidated method to augment the training set in imbalanced problems. A synthetic feature vector can obtained from every original feature vector by random interpolation with some selected neighbors of the original training set. Finally, the dot-dash line curve corresponds to data augmentation using the method GANSO described in the previous section. This method requires a specific design for every particular application. Most importantly, structural information of the feature vector must be defined to obtain more realistic replicas, as indicated in Section 3. To this aim, every 10-dimensional feature vector **f** is partitioned into 4 blocks by grouping the features corresponding to the same type of sensor, namely

$$\begin{aligned}&\mathbf{f}=\left[\mathbf{f}_A^T,\mathbf{f}_M^T,\mathbf{f}_S^T,\mathbf{f}_W^T\right]^T\\&\mathbf{f}_A=[f_{A1},f_{A2},f_{A3}]^T;\ \mathbf{f}_M=[f_{M1},f_{M2},f_{M3}]^T\\&\mathbf{f}_S=[f_{S1},f_{S2}]^T;\ \mathbf{f}_W=[f_{W1},f_{W2}]^T\end{aligned}\qquad(1)$$

where A stands for "accelerometer" (3 sensors), M stands for "microphone" (3 sensors), S stands for "speed" (2 sensors), and W stands for "wheel" (2 sensors). This partitioning leads to the graph model required by GANSO. A graph of 10 nodes (one for every feature) is built, where features of the same type of sensor are connected while features from different types of sensors are disconnected. This represents, in a rather simple manner, the assumed structural information to be incorporated into the synthesis procedure, i.e., features from the same types of sensors are assumed to exhibit some degree of dependence if compared with features from different types of sensors. Let us briefly explain the main steps of GANSO in this specific application. Figure 4 describes the essential concepts. We start from a set of real feature vectors available for training $F\ :\ \left\{\mathbf{f}^{(i)},\ i=1\ldots TSS\right\}$; this is the reference set in Figures 2 and 4. From every element of set F, we want to generate a synthetic replica so that the total number of training feature vectors is duplicated. Let us consider an arbitrary member **f** of F (OFV in Figures 2 and 4). Using the surrogating procedure described in [31] with the above-defined graph model, the generator block of Figure 2 provides a synthetic candidate $\mathbf{f}^s$ (SFV in Figures 2 and 4) to the discriminator block. Then, two sets of features are computed by, respectively, comparing the real feature vector **f** and the synthetic candidate $\mathbf{f}^s$ with every element $\mathbf{f}^{(i)}$ of F. Comparison is made by computing structured correlation, which means that actually 4 correlations are computed, one for every partition as defined in (1), thus obtaining 4-dimensional feature vectors $\mathbf{s}_i$ and $\mathbf{s}_i^S$:

$$\mathbf{s}_i\left(\mathbf{f},\mathbf{f}^{(i)}\right)=\begin{bmatrix}corr\left(\mathbf{f}_A,\mathbf{f}_A^{(i)}\right)\\corr\left(\mathbf{f}_M,\mathbf{f}_M^{(i)}\right)\\corr\left(\mathbf{f}_S,\mathbf{f}_S^{(i)}\right)\\corr\left(\mathbf{f}_W,\mathbf{f}_W^{(i)}\right)\end{bmatrix}\quad\mathbf{s}_i^S\left(\mathbf{f}^S,\mathbf{f}^{(i)}\right)=\begin{bmatrix}corr\left(\mathbf{f}_A^s,\mathbf{f}_A^{(i)}\right)\\corr\left(\mathbf{f}_M^s,\mathbf{f}_M^{(i)}\right)\\corr\left(\mathbf{f}_S^s,\mathbf{f}_S^{(i)}\right)\\corr\left(\mathbf{f}_W^s,\mathbf{f}_W^{(i)}\right)\end{bmatrix}\qquad(2)$$
$$i=1\ldots TSS$$

where $corr(\mathbf{v},\mathbf{w})=\frac{\mathbf{v}^T\mathbf{w}}{\|\mathbf{v}\|\|\mathbf{w}\|}$ is the normalized correlation between the two vectors **v** and **w** and $\|\cdot\|$ stands for the Euclidean norm. Thus, two sets of size TSS of 4-dimensional feature vectors are obtained. Let us assume that every set corresponds to a different class of a two-class problem. Then, the discriminator block implements a linear discriminant in an effort to separate these two classes. Let us call **d** the linear discriminant vector learned from the two mentioned sets. Assuming that the overall mean vector has been extracted from the two sets, perfect separability would imply that

$$\mathbf{d}^T\mathbf{s}_i>0\qquad\mathbf{d}^T\mathbf{s}_i^S<0\qquad i=1\ldots TSS\qquad(3)$$

In practice, perfect separability is not possible, but we can impose a less strict condition to decide if both classes are separable or not. For example, we could decide that both classes are non-separable if (3) is not satisfied in more than half of the cases. If both classes turn out to be non-separable, the candidate is accepted to augment the set of real feature vectors. Otherwise, the candidate is rejected, and a new one must be provided by the generative block. To this aim, the generator is "informed" by the discriminator about the learned discriminant **d**, and then the generator changes the signs of the partitions of $\mathbf{f}^S$ contributing to a decrease in $\mathbf{d}^T\mathbf{s}_i^S \;\; i = 1\ldots TSS$ in more than $TSS/2$ cases, in an effort to reduce the number of cases in (3) in which $\mathbf{d}^T\mathbf{s}_i^S < 0$, thus decrementing the separability between both sets. The modified candidate is provided to the discriminator and a new iteration is repeated until the candidate is finally accepted. In our experiments, two or three iterations have been enough most of the time.

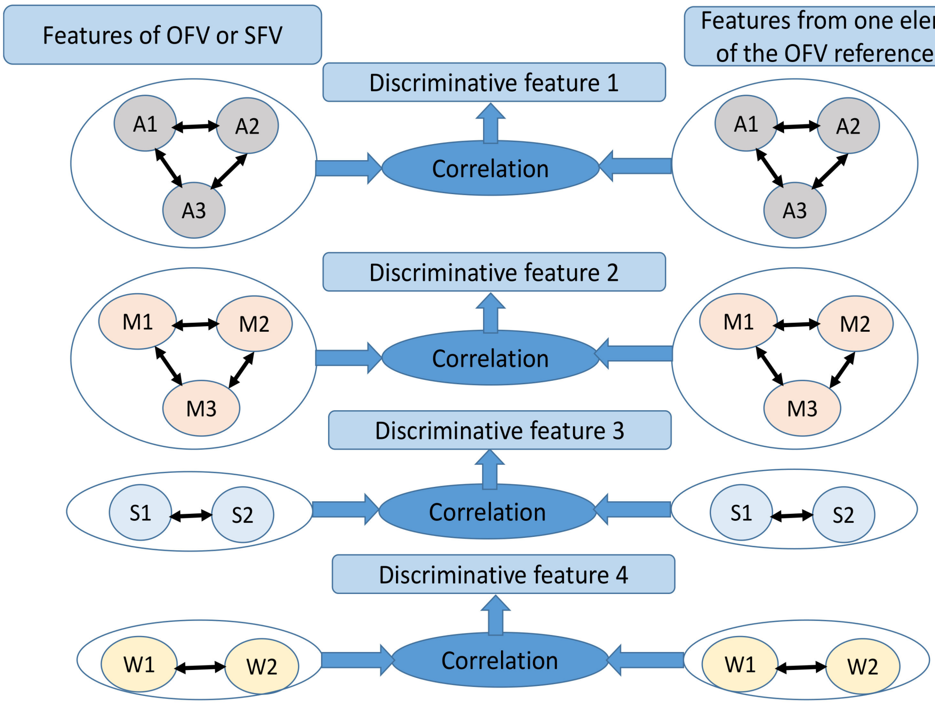

Figure 4. Graph model and structured correlation between OFV or SFV and the OFV reference set. Sensor acronyms refer to Figure 1 nomenclature: A, accelerometer; M, microphone; S, speed; W, wheel.

The curves in Figure 3 show that, in all cases, accuracy improves by increasing *TSS*. Notice that for a given *TSS*, the accuracy improves both in LDA and RDF by using GANSO for data augmentation, with respect to not using data augmentation. However, data augmentation with SMOTE does not show a significant improvement, neither with LDA nor with RDF. Note that SMOTE is a method normally used for imbalance problems, where only the minority class is oversampled, but, here, we consider a problem where all classes are oversampled. Moreover, incorporating structural information seems to be crucial to obtain realistic replicas of the real feature vectors. On the other hand, as expected, RDF achieves higher accuracy than LDA both with and without data augmentation. However, the complexity of training, the implementation, and the overfitting problems make LDA a better option.

Notice that the improvement with GANSO decreases as *TSS* increases; it is because, for large *TSS*, we are approximating the Bayes error rate (*BER*) [36,37]. *BER* is the probability of error (P_e) that could be achieved if we could have perfect knowledge of the class conditional probability densities in the feature space. In a practical setting, we never have this knowledge; instead, we have a finite training dataset. Thus, *BER* cannot be achieved but, assuming consistent estimates of the model, *BER* can be approximated as *TSS* increases to infinity. Remember that, in Figure 3, we showed the accuracy, i.e., $(1 - P_e)$.100; thus, all curves converge to (1-*BER*).100 for increasing *TSS*. However, this limit is reached before with data augmentation by GANSO.

For a better quantification of the improvement provided by data augmentation, we have defined the concept of effective training set size (*ETSS*), which is the *TSS* with no data augmentation (NDA) that would be required to achieve the same accuracy obtained by a given TSS with data augmentation (DA). This can be expressed as:

$$ETSS = TSS + \alpha \cdot N_{DA} \tag{4}$$

where N_{DA} is the number of feature vectors added to the TS and α is a factor due to DA. Thus, with NDA ($N_{DA} = 0$), it is $ETSS = TSS$. As indicated, in our DA experiments, we duplicate TSS so that $N_{DA} = TSS$ and (3) becomes

$$ETSS = TSS \cdot (1 + \alpha) \tag{5}$$

Thus, Figure 5 shows $ETSS$ as a function of TSS with NDA and with DA. With NDA, $ETSS$ coincides with TSS ($\alpha = 0$). For DA, we have considered the two types of classifiers (LDA and RDF) and the method GANSO (SMOTE cannot provide a significant increase in the $ETSS$). Notice in Figure 5 that $ETTS$ is always above TSS ($\alpha > 0$). This implies that DA allows a reduction by a factor $(1 + \alpha)$ in the actual TSS required to achieve a target accuracy. For example, from Figure 3, we can see that TSS = 1400 with NDA achieves an accuracy of 91%, but this same accuracy can be obtained for TSS = 880 with LDA-GANSO, i.e., a reduction factor $1400/880 = 1.59$. Actually, the mean reduction factor achieved by LDA-GANSO in the TSS interval of Figure 5 is 1.52.

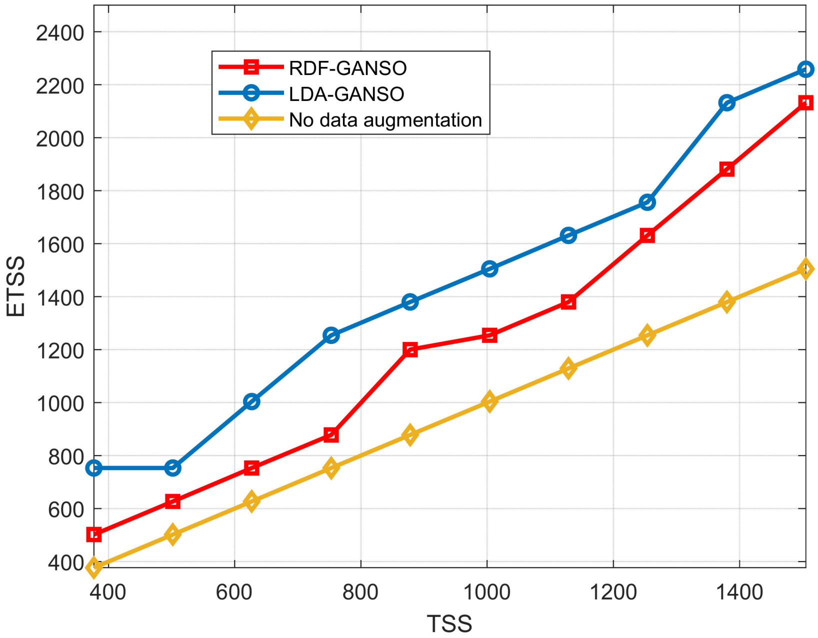

Figure 5. Effective training set size ($ETSS$) for a given training set size (TSS), with data augmentation (RDF-GANSO, LDA-GANSO) and with no data augmentation.

5. Discussion and Conclusions

It has been demonstrated that data augmentation is a promising approach to reduce the size of the captured dataset required for training road surface classifiers for automatic power-assisted steering. In the experiments included in this work, a reduction by an approximate factor of 1.5 has been shown when using the LDA classifier and GANSO for data augmentation. The reduction factor of TSS directly applies to the reduction of the signal acquisition time and of the number of captures. However, it is hard to quantify the impact of the TSS reduction on the overall time saving of the whole capture experiment. Several other factors, such as the number of different configurations and the time required to change the configuration setting, have to be considered, among others.

Several issues are open in any of the basic blocks of the multisensory road surface classification system regarding the actual reduction that could be achieved. Thus, for example, a different accuracy is obtained when using RDF instead of LDA, and other types of classifier could be tested. In any case, notice that the aim of this communication is to show that data augmentation can provide significant simplification of the tedious capture of real data, rather than to compare different options of the classification method. Definitely, given the practical focus of the work, LDA is the most attractive alternative.

Other elements of the approach deserve further research. For example, the sensor configuration has a variety of possibilities. In particular, the inclusion of sensors (e.g., microphones) that are not standard elements of the car is an issue. Furthermore, other features could be tried, as well as different procedures for feature selection. Finally, different strategies of data augmentation are possible besides duplicating the original *TS*.

In conclusion, the presented communication motivates future research in both theoretical and experimental aspects of road surface classification to aid the driving assistance systems of the car by automatic power-assisted steering. The presented results should encourage researchers to keep on working on data augmentation to ease the tedious and costly capture of the training dataset.

Author Contributions: Conceptualization, A.S., A.R. and L.V.; methodology, A.S., A.R. and L.V.; software, A.S and N.V.; validation, A.S., A.R. and L.V.; formal analysis, A.S., A.R. and L.V.; investigation, A.S., A.R. and L.V.; resources, A.R.; data curation, N.V.; writing—original draft preparation, A.S. and L.V.; writing—review and editing, A.S., N.V. and L.V.; supervision, L.V.; project administration, L.V.; funding acquisition, L.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by MCIN/AEI/10.13039/501100011033 and by the European Union, grant number TEC2017-84743-P.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tudón-Martínez, J.C.; Fergani, S.; Sename, O.; Martinez, J.J.; Morales-Menendez, R.; Dugard, L. Adaptive road profile estimation in semiactive car suspensions. *IEEE Trans. Control. Syst. Technol.* **2015**, *23*, 2293–2305. [CrossRef]
2. Prażnowski, K.; Mamala, J. Classification of the road surface condition on the basis of vibrations of the sprung mass in a passenger car. *IOP Conference Series: Materials Science and Engineering* **2016**, *Volume 148*, 012022. [CrossRef]
3. Qin, Y.; Langari, R.; Wang, Z.; Xiang, C.; Dong, M. Road excitation classification for semi-active suspension system with deep neural networks. *J. Intell. Fuzzy Syst.* **2017**, *33*, 1907–1918. [CrossRef]
4. Gueta, L.B.; Sato, A. Classifying road surface conditions using vibration signals. In Proceedings of the 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, Malaysia, 12–15 December 2017; pp. 39–43.
5. Bystrov, A.; Hoare, E.; Tran, T.Y.; Clarke, N.; Gashinova, M.; Cherniakov, M. Automotive surface identification system. In Proceedings of the 2017 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Vienna, Austria, 27–28 June 2017; pp. 115–120.
6. Surblys, V.; Žuraulis, V.; Sokolovskij, E. Estimation of road roughness from data of on-vehicle mounted sensors. *Eksploat. I Niezawodn.-Maint. Reliab.* **2017**, *19*, 369–374. [CrossRef]
7. Park, J.; Min, K.; Kim, H.; Lee, W.; Cho, G.; Huh, K. Road surface classification using a deep ensemble network with sensor feature selection. *Sensors* **2018**, *18*, 4342. [CrossRef] [PubMed]
8. Han, K.; Choi, M.; Choi, S.B. Estimation of the tire cornering stiffness as a road surface classification indicator using understeering characteristics. *IEEE Trans. Veh. Technol.* **2018**, *67*, 6851–6860. [CrossRef]
9. Bystrov, A.; Hoare, E.; Tran, T.-Y.; Clarke, N.; Gashinova, M.; Cherniakov, M. Sensors for Automotive Remote Road Surface Classification. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, 12–14 September 2018.
10. Yusoff, S.M.; Giacomin, J. The effect of vibrational energy distribution on the level of driver detection. *AIP Conf. Proc.* **2019**, *2059*, 020032.

11. Ng, J.R.; Wong, J.S.; Goh, V.T.; Yap, W.J.; Yap, T.T.V.; Ng, H. Identification of Road Surface Conditions using IoT Sensors and Machine Learning. In *Computational Science and Technology*; Springer: Singapore, 2019; pp. 259–268.
12. Beilfuss, T.; Kortmann, K.-P.; Wielitzka, M.; Hansen, C.; Ortmaier, T. Real time classification of road type and condition in passenger vehicles. *IFAC Pap.* **2020**, *53*, 14254–14260. [CrossRef]
13. Safont, G.; Salazar, A.; Rodríguez, A.; Vergara, L. Comparison of Dimensionality Reduction Methods for Road Surface Identification System. In Proceedings of the 2020 Science and Information Conference, London, UK, 16–17 July 2020; pp. 554–563.
14. Safont, G.; Salazar, A.; Rodriguez, A.; Vergara, L. Multichannel Signal Processing for Road Surface Identification. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3052–3056.
15. Chugh, T.; Bruzelius, F.; Klomp, M.; Jacobson, B. Steering feedback transparency using rack force observer. *IEEE/ASME Trans. Mechatron.* **2022**, in press. [CrossRef]
16. Bonera, E.; Gadola, M.; Chindamo, D.; Morbioli, S.; Magri, P. On the influence of suspension geometry on steering feedback. *Appl. Sci.* **2020**, *10*, 4297. [CrossRef]
17. Yaohua, L.; Jikang, F.; Jie, H.; Youfei, N.; Qianlong, F. Novel electric power steering control strategies of commercial vehicles considering adhesion coefficient. *Adv. Mech. Eng.* **2020**, *12*. [CrossRef]
18. Raudys, S.; Duin, R.P.W. Expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix. *Pattern Recognit. Lett.* **1998**, *19*, 385–392. [CrossRef]
19. Berikov, V.B. An approach to the evaluation of the performance of a discrete classifier. *Pattern Recognit. Lett.* **2002**, *23*, 227–233. [CrossRef]
20. Berikov, V.B.; Litvinenko, A. The influence of prior knowledge on the expected performance of a classifier. *Pattern Recognit. Lett.* **2003**, *24*, 2537–2548. [CrossRef]
21. Rueda, L. A one-dimensional analysis for the probability of error of linear classifiers for normally distributed classes. *Pattern Recognit.* **2005**, *38*, 1197–1207. [CrossRef]
22. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
23. He, H.; García, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
24. Chi, Y.C.; Wang, H. Establish a patent risk prediction model for emerging technologies using deep learning and data augmentation. *Adv. Eng. Inform.* **2022**, *52*, 101509. [CrossRef]
25. Wali, A.; Alamgir, Z.; Karim, S.; Fawaz, A.; Ali, M.B.; Adan, M.; Mujtaba, M. Generative adversarial networks for speech processing: A review. *Comput. Speech Lang.* **2022**, *72*, 101308. [CrossRef]
26. Abdou, M.A. Literature review: Efficient deep neural networks techniques for medical image analysis. *Neural Comput. Appl.* **2022**, in press. [CrossRef]
27. Setiawan, B.D.; Serdült, U.; Kryssanov, V. A machine learning framework for balancing training sets of sensor sequential data streams. *Sensors* **2021**, *21*, 6892. [CrossRef]
28. Setiawan, B.D.; Serdült, U.I.; Kryssanov, V. Smartphone sensor data augmentation for automatic road surface assessment using a small training dataset. In Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea, 17–20 January 2021; pp. 239–245.
29. Choi, D.G. Image based road surface classification method using CNN. *Int. J. Recent Technol. Eng.* **2019**, *8*, 158–162.
30. Salazar, A.; Vergara, L.; Safont, G. Generative Adversarial Networks and Markov Random Fields for oversampling very small training sets. *Expert Syst. Appl.* **2021**, *163*, 113819. [CrossRef]
31. Belda, J.; Vergara, L.; Salazar, A.; Safont, G.; Parcheta, Z. A new surrogating algorithm by the complex graph Fourier transform (CGFT). *Entropy* **2019**, *21*, 759. [CrossRef]
32. Belda, J.; Vergara, L.; Salazar, A.; Safont, G. Estimating the Laplacian matrix of Gaussian mixtures for signal processing on graphs. *Signal Process.* **2018**, *148*, 241–249. [CrossRef]
33. Peeters, G. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO IST Proj. Rep.* **2004**, *54*, 1–25.
34. Duda, R.O.; Hart, P.E.; Stork, D.H. *Pattern Classification*, 2nd ed.; Wiley Interscience: Hoboken, NJ, USA, 2000.
35. Ho, T.K. The Random Subspace Method for Constructing Decision Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844.
36. Nielsen, F. Generalized Bhattacharyya and Chernoff upper bounds on Bayes error using quasi-arithmetic means. *Pattern Recognit. Lett.* **2014**, *42*, 25–34. [CrossRef]
37. El Ayadi, M.M.H.; Kamel, M.S.; Karray, F. Toward a tight upper bound for the error probability of the binary gaussian classification problem. *Pattern Recognit.* **2008**, *41*, 2120–2132. [CrossRef]