Check for updates

DOI: 10.1111/mice.13014

INDUSTRIAL APPLICATION

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Efficient data dimensionality reduction method for improving road crack classification algorithms

Francisco J. Rodriguez-Lozano | **Juan C. Gámez-Granados** | **Jose M. Palomares** | **Joaquín Olivares**

Department of Electronic and Computer Engineering, Universidad de Córdoba, Córdoba, Spain

Correspondence
Joaquín Olivares, Department of Electronic and Computer Engineering, Universidad de Córdoba, Edif. Leonardo da Vinci, C.U. Rabanales, Córdoba, 14071, Spain.
Email: olivares@uco.es

Funding information
VISNAV, Grant/Award Number: RFFINNL-298890; Norwegian Research Council; Advanced Informatics Research Group, Grant/Award Number: TIC-252

Abstract

Automatic crack classification plays an essential role in road maintenance. Using many features for the classification is inefficient for implementing embedded systems with low computational resources makes it difficult. Therefore, this work proposes a new data dimensionality reduction (DDR) for crack classification algorithms (DDR4CC). DDR4CC reduces the required information about the cracks to only four features. Using these features, the images can be classified into longitudinal, transverse, and alligator cracks or healthy pavement. DDR4CC is compared with eight DDR methods, and the reduced set of features is analyzed using five different classification algorithms. Besides, five different datasets, generated by a combination of several public datasets, are used. We are proposing a simple DDR method with high interpretability of the data, obtaining very fast computation and high accuracy. Experiments show that DDR4CC enhances the results of the classification algorithms, providing almost perfect classifiers with a minimum computation time.

1 | INTRODUCTION

An optimal status of the road surface reduces maintenance costs. So, one of the early tasks in road health monitoring is the inspection of the road surface, specifically, the search for cracks. Traditionally, this activity has been achieved manually, causing this task to be expensive, tedious, and inefficient. To mitigate this problem, machine learning algorithms along with computer vision methods have been suggested. However, automatic approaches to road surface condition inspection are not straightforward, and therefore the problem is divided into two specific tasks as Figure 1 shows: crack detection and crack classification.

Approaches based on crack detection (Chu et al., 2022) usually have two different objectives. The first objective is to obtain the regions of interest where defects are located. The second one is the creation of a binary image (typical values between 0 and 255) that emphasizes the pixels that correspond to the damaged surface (Y. Zhang & Yuen 2021). These proposals range from algorithms based on image enhancement (Hoang, 2018), illumination correction (C. Chen et al., 2021), and edge detection (Cubero-Fernandez et al., 2017) to others based on more sophisticated methods, such as minimal path estimation (Amhaz et al., 2016), shadow removal (Palomar et al., 2010), or even classic machine learning algorithms (Y. Shi et al., 2016). Furthermore, the emerging growth of deep learning algorithms is being used widely, and we will review it in Section 2.

Approaches based on crack classification need to perform the detection task as well, but despite this fact, the number of proposals is lower than for solely crack

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2023 The Authors. *Computer-Aided Civil and Infrastructure Engineering* published by Wiley Periodicals LLC on behalf of Editor.

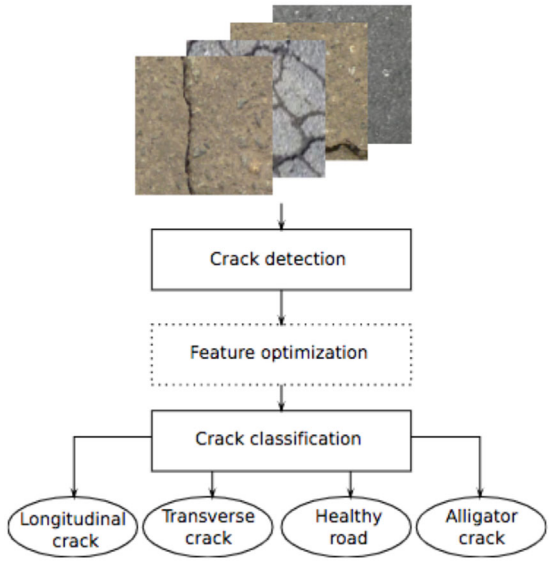

FIGURE 1 Tasks for road crack classification and the new optimization feature task.

detection ones (Hamishebahar et al., 2022). Besides, there is no consensus on the number or the name of the classes.

For instance, there are proposals focused on a binary classification to detect whether there is a crack or not (Mokhtari et al., 2017). Other proposals have been published centered on the classification of a single specific type of cracks, such as potholes (Jana et al., 2022), longitudinal and transverse cracks (Ibrahim et al., 2019), classification of lineal cracks (Liang et al., 2018), or even in the classification of raveling severity (Tsai et al., 2020). However, despite the different types into which defects can be classified, three basic classes can be found that are precursors of later defects (Garber & Hoel 2008): alligator cracks, transverse cracks, and longitudinal cracks. These classes were widely used by different authors employing mixed image processing and machine learning approaches (Cubero-Fernandez et al., 2017; Hoang & Nguyen 2018; Li et al., 2021; Osman et al., 2021; Rodriguez-Lozano et al., 2020).

In addition, most studies based on crack classification usually employ a very large number of features that are used as the input data for the machine learning algorithms to learn the relationships between the data. This large number of features has a large impact on the results of the subsequent models.

The problem of high dimensionality data in civil engineering problems was introduced by Adeli and Wu (1998). On the one hand, it has negative consequences on the accuracy of the classification algorithms due to a large number of features, which leads to the generation of complex models that tend to misclassify the different images into classes. On the other hand, the high number of features has an impact on the time used to carry out the classification because when generating more complex models, more operations must be performed by the models to reach the decision-making. This effect is also amplified when we are dealing with embedded devices that have limited computational capacity and memory.

We propose the crack classification method for data dimensionality reduction (DDR4CC). DDR4CC introduces a new feature optimization task placed between the detection and classification tasks as shown in Figure 1. DDR4CC boosts classification accuracy while consuming less time. It also provides interpretable data and can be used with images of any resolution.

This paper is organized as follows: A description of related works is shown in Section 2. Some brief explanations of different DDR approaches are described in Section 3. Section 4 contains the specifications of several machine learning algorithms. Section 5 shows our DDR proposal. The five datasets used for experimentation are described in Section 6. The results are analyzed and discussed in Section 7, demonstrating that our DDR method takes advantage of the data interpretability obtaining high accuracy with low computing. Finally, the main conclusions are presented in Section 8.

2 | RELATED WORKS

Recently, a huge amount of data is obtained in different areas of application, resulting in an exponential growth in complexity, heterogeneity, dimensionality, and size. The classical statistical methodologies from an era where the collection of data was complex, and the magnitude of datasets was much smaller, are not appropriate (Nanga et al., 2021). So, there is a challenge in analyzing these large and sophisticated datasets. DDR is applied when data with vast dimensions are reduced to lesser dimensions but ensures that it concisely conveys similar information. DDR can be performed through feature selection and feature extraction. DDR as a result facilitates the classification, visualization, and compression of high-dimensional data. Furthermore, the DDR is used in neural dynamic classification algorithms (Rafiei & Adeli 2017b), applied in different problems, such as earthquake early earning detection (Rafiei & Adeli 2017a), or to detect damage in highrise building structures (Rafiei & Adeli 2017c).

As mentioned in Section 1, deep learning (DL) techniques are applied for classification. One of the main drawbacks of the DL techniques is their lack of interpretability of decision-making. DL techniques are designed as black boxes, impeding the knowledge of the inner behavior. Moreover, the hardware requirements these techniques require are really high, preventing these techniques to be

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

executed in real low-end devices, such as the ones that would be included in maintenance vehicles.

Besides the abovementioned loss of interpretability of the DL methods, most recent crack classification works based on lightweight DL are analyzed, in order to state the quality of the classification (in terms of F1-score metric) and the speed regarding the processing unit used for the computation. It is to be noted that most of the works in recent years are focused on crack detection and not on crack classification. Crack detection is a step to be taken in a previous step in crack classification; however, they require different models. Several recent proposals based on DL were presented, however, mostly for concrete surfaces (Çelik & König 2022; Chun et al., 2021; Jang et al., 2021; Kong et al., 2021; Liu & Xu 2022; Xie et al., 2022; Zheng et al., 2022). Regarding road or pavement cracks, an image-to-image translation was presented for night images in Liu and Xu (2022), and finally J. Chen and He (2022) presented a neural network (NN) for the detection of four types of pavement cracks.

In the work of Hou et al. (2022), three different methods are tested: Faster region-based convolutional neural networks (R-CNN), YOLOv3, and fast shape-based network (FS-Net). The authors use 4K images. The timings are 794 ms for Faster R-CNN, 81 ms for YOLOv3, and 83 ms for FS-Net. The precision metric ranges from 0.894 to 0.943, which is quite high. However, the recall metric is extremely low for all the models (0.0132, 0.0113, and 0.0074). Thus, the overall quality (in terms of F1-score to be able to compare them with those in our work) is very low. Moreover, even if the timing is fast, it is to be mentioned that those processing times are obtained using a very powerful structure: an Intel Xeon Gold 6151 at 3.0GHz with 128 GB RAM and with a NVIDIA TITAN RTX 24GB GPU. The hardware to be used in a maintenance vehicle should be much more lightweight.

In Espindola et al. (2022) a large review of different lightweight DL models for crack detection and classification was presented. It is stated that a DL network's computational load requirement is the number of floating-point operations per second (FLOPS). Computational loads of several classifier networks are provided: ResNet-34, ResNet-50, and The VGG model stands for the Visual Geometry Group from Oxford (VGG16) range from 4 to 15.5 giga-floating point operations per second (GFLOPS). The detection using YOLOv3-Darknet53 requires 65 GFLOPS and segmentation with U-Net using 221 GFLOPS. Furthermore, our proposal does not require floating-point operations but integer operations, which are much more efficient. For example, DDR4CC using the typical 640 × 480 resolution (larger than usual DL methods, which use 300 × 300 or 256 × 256) obtains the four features reduction in only 0.1202 s, which means 2.5 millions of instructions per second (MIPS). If we include the Best First Tree (BFTree) classification model, the complete timing (both reduction and classification) is 0.2042 s or 1.5 MIPS. Note that GFLOPS and MIPS are not directly comparable; however, we could estimate this value as more than 1000 smaller than those from lightweight DL models, providing similar results in terms of classification performance.

Another recent work is Wan et al. (2022). In it, a lightweight DL you only look once - lightweight method for road damage detection (YOLO-LRDD) is used and compared to Faster R-CNN. YOLO-LRDD is improved from the YOLOv5s. YOLO-LRDD takes 11.63 ms per image, while Faster R-CNN takes 19.61 ms. They are fast; however, the quality is low: Faster R-CNN scores 0.601, which is higher than YOLO-LRDD with 0.587. Moreover, the processing times are obtained with a powerful device (NVIDIA RTX 3060 w/12GB).

A very interesting work is done by (Ali et al., 2022). In it, they have provided a large survey on DL based on different CNN for crack detection (although those cracks are not classified). They analyze a large number of articles focusing on their quality. They provide F1-score, ranging from 0.489 to 0.9967. They also include the hardware used in each article (if provided). In all of those articles that declare the hardware, it is very more powerful. Therefore, the processing requirements for DL are much higher than those required in this work.

Current DL techniques are improving both in quality and speed processing. They are optimized for lightweight processing. However, most of the lightweight DL cannot be executed in real low-end devices such as NVIDIA Jetson Nano or similar, which would be the one used in maintenance vehicles. Thus, the use of non-DL methods is clearly justified for computing processing requirements and for the interpretability of decision-making.

Therefore, focusing on simple approaches with DDR techniques, Ibrahim et al. (2019) carry out a study of longitudinal and transverse cracks by applying computer vision algorithms to extract the binary image and a data transformation before feeding two different versions of the "K nearest neighbor (KNN)" algorithm. The new features to feed the KNN classifiers are the length of X and Y axes.

The work proposed by Sheerin et al. (2018) implements Otsu's method for several works focused on the detection and classification of pavement cracks. They suggest a system that makes use of the wavelet transform and singular value decomposition (SVD) to extract and reduce the main features to perform the classification.

Ahmadi et al. (2018) extract from binary images the number of rows, columns, standard deviations, and the number of pixels that represents a crack, and then use principal component analysis (PCA) to reduce the number of features.

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Finally, theybmake a comparison applying KNN, support vector machines (SVM), and decision trees (DT).

Also, Abdel-Qader et al. (2006) apply three different approaches to PCA to reduce the number of features in bridge crack detection applying: raw image data, a linear structure detector, and, PCA in conjunction with local principal components, to segment the images in small tiles of 16×16 blocks.

C. Chen et al. (2021) use an image processing method based on local binary patterns (LBP) to feed an SVM classifier to classify pavement cracks into transverse, longitudinal, alligator, and non-crack classes. Furthermore, PCA is used to reduce the amount of LBP features. The best results are 76.01% for alligator cracks.

Kusumaningrum et al. (2022) propose a semi-automatic image processing method for the classification of pavement defects into cracks, potholes, and alligator cracks. Then, they apply histogram and morphological operations to extract a binary image representing pavement defects. They simplify the data by performing a radial vector positioned at the centroid of the largest defect area, and the eight cardinal directions are used for the analysis of the ratio of positive and negative pixels along the line in each angle direction. These radial vectors are the data used as representative of the defects in a KNN classifier. It provides 83% accuracy at the highest value.

Ahmadi et al. (2021) compute binary images, apply the Hough transform, and calculate some angle adjustments to obtain 24 features divided into three sets to define the different cracks. Then, different classifiers are used in several stages, using bagged trees to differentiate between longitudinal, transverse, and diagonal/block cracks and using an SVM classifier to differentiate between diagonal and block cracks. In addition, they compare different models such as KNN, NNs, DT, and those described above, obtaining 93.86% overall accuracy.

Xu et al. (2018) classify defects into four types: distress-free, cracking, pothole, and patching. They obtain four features from the texture using the linear discriminant analysis (LDA) method. Also obtain three features from the shape of the binary images: the average area of all connected components, the area of the maximal connected component, and the equivalent length of the longest connected component.

Finally, Cubero-Fernandez et al. (2017) and Rodriguez-Lozano et al. (2020) obtain a binary image. Then, a method known as projections (Gonzalez & Woods 2018) is applied in order to reduce the number of features of the whole image to just the number of features in each column and row of each image. This work focuses on the classification of the same classes as the current work using an ensemble of models. Also, the authors compare the obtained results with other algorithms and methods, providing an average result of 91.8% and 91.5% according to precision and recall metrics.

Once the proposals to reduce the number of features have been reviewed, it is possible to observe that, in general, there are certain limitations:

1. Classification methods require the resolution (size) of the image to be changed to feed the trained models.
2. Some of the features selected can be based on the physical location of the crack in the image. Therefore, a change in the position of the crack may result in misclassification.
3. There is no information about any set of features that performs accurately in several different machine learning algorithms.
4. The set of features proposed in most works is not straightforward to understand or calculate.

Our work overcomes all the limitations presented above.

3 | PRELIMINARIES IN DDR METHODS

This section provides a light introduction to the DDR methods used by other authors. Each of them will be compared with our proposal in Section 7.

LDA (Mai, 2013) is a method used for dimensionality reduction as well as for classification problems, by providing class separability with a decision region between the different classes. The aim of LDA is to find a linear projection for data that maximizes the variance between classes relative to the variance for data from the same class.

PCA (Jolliffe & Cadima 2016) is used to find patterns in the data of a dataset. This method is frequently employed for data compression. These patterns are discovered by applying statistical and algebraic operations to the data. The result is a list of features that summarize a large amount of data. It is common to select a subset of features to describe a certain percentage of the variance of a dataset, for instance, 95% of the variance of the dataset.

PCA-4 PCA provides a list of features from which a subset is usually selected to explain a given percentage of the variance of the dataset. A different approach is to provide a sorted list of features, ordered by the amount of percentage of the variance of the dataset explained by that feature, and select the first features of that list. The amount of features is a parameter provided by the user. In this case, the amount of features selected from the ordered list is four.

SVD (Dhumal & Deshmukh 2016) achieves dimension reduction through matrix decomposition without needing to calculate the covariance matrix.

t-Distributed stochastic neighbor embedding (TSNE; van der Maaten & Hinton 2008) captures non-linear

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

relationships in the data and uses conditional probabilities of similarity between points instead of Euclidean distance. It constructs a probability distribution over pairs of samples in the original space so that similar samples receive a high probability of being chosen; meanwhile, very different samples receive a low probability of being chosen.

Multidimensional scaling (MDS; Hout et al., 2013) obtains the dimensionality reduction by finding underlying attributes or dimensions with more influence. It models the similarity of data by calculating distances (D) between each pair of points.

Scalable algorithm for capturing local and global features of high-dimensional datasets (IVIS; Szubert et al., 2019) is a framework for dimensionality reduction of single-cell expression data that uses a Siamese NN (SNN) architecture. The IVIS SNN is composed of three identical base networks with three layers of 128 neurons and the final embedding layer.

Principal projections (PP). The use of projective integrals in the works of (Cubero-Fernandez et al., 2017; Hoang, 2018; Rodriguez-Lozano et al., 2020) allowed the reduction of the data required to provide a good classification of the cracks in a fast mode. These authors proposed a special use of the projective integrals, in which the amount of edge-detected pixels is aggregated for every row and column of the edge-processed images. Thus, the amount of features required for the classification is reduced from (width × height) to (width + height).

All these methods have been proposed to reduce the number of required data, for further processing, usually classification. The PP method is the only described method specifically designed for images. All the rest are generic and may be used with any data scope. All these methods will be tested against our DDR proposal.

4 | PRELIMINARIES IN MACHINE LEARNING ALGORITHMS

This is a brief introduction to the mechanics of some shallow machine learning algorithms that will be used to test our DDR proposal explained in Section 5. These algorithms have been selected because they cover both simple and complex approaches in their procedures to classify patterns:

Naïve Bayes (H. Zhang & Jiang 2022) is a probabilistic classification algorithm that focuses on data labeling by extracting statistical information from the available data of the training set.

The KNN (Gou et al., 2022) algorithm labels new patterns based on the distance to existing training data. Thus, to classify new data, it is necessary to explore all the training data. As a metric distance, the Euclidean distance is the most popular one, but other metrics may be used (Alfeilat et al., 2019).

BFTree (H. Shi, 2007) uses a "divide-and-conquer" approach to generate a classification tree with binary partitions. In this tree, the *nodes* are the attributes that separate the instances in the different classes, the *branches* are the decision rules to take a path, and the leaves correspond to the final labels of the data.

In this method, each division of the tree into nodes is made by taking the best attribute, which is able to divide the patterns using the impurity metric.

Partial decision trees (PART; Ibarguren et al., 2016) generate rules using a tree scheme following a "divide-and-conquer" approach. This method was developed as an efficient approach, compared to C4.5 (Quinlan, 1993) and RIPPER (Cohen, 1995), avoiding complex optimization and stages or adjustments to modify the individual rules of the ruleset. In contrast, this method generates the rules using partial trees, which are then pruned. The metric, *Entropy*, used to split the tree into branches is the same as in the C4.5 algorithm.

SVM (Chang & Lin 2011) is used for classification or regression purposes. The aim of the SVM is to find a hyperplane that separates the data linearly into a positive class and a negative class.

These are the most usual methods used in machine learning for classification tasks. In this work, all these methods are tested in combination with the DDR methods described in Section 3 and our proposal.

5 | DDR FOR CRACK CLASSIFICATION ALGORITHMS (DDR4CC)

DDR4CC uses as input binary images obtained from a binarization method (for instance, Chu et al., 2022; Cubero-Fernandez et al., 2017; Hoang, 2018; Meng et al., 2022). Binary images usually contain a great number of 0 values. These values are features themselves; however, most of them are not useful for the classification step, making the machine learning models unnecessarily complex. Hence, it is necessary to reduce the number of features. For this reason, the first step of the DDR4CC is to compress the information of the binary image avoiding all 0 values. This compression is performed by obtaining the cumulative sum of the column and row pixel values by applying Equations (1) and (2):

$$Row^{Acc}\left(I_j\right) = \frac{\sum_{i=1}^{C-1} I\left(i,j\right)}{255 \cdot R}; j \in [1,R] \tag{1}$$

$$Colm^{Acc(I_i)} = \frac{\sum_{i=1}^{R-1} I\left(i,j\right)}{255 \cdot \mathrm{C}}; i \in [1,C] \tag{2}$$

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

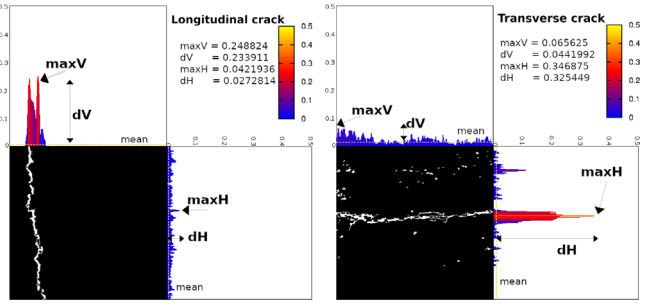

FIGURE 2 Example of accumulated pixels and the newly acquired features for longitudinal and transverse classes.

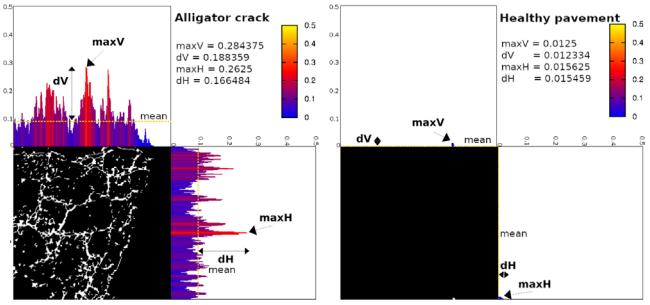

FIGURE 3 Example of accumulated pixels and the newly acquired features for alligator and healthy pavement classes.

Let $Row^{Acc}(I_j)$ and $Colm^{Acc}(I_i)$ be the vectors of accumulated values of rows and columns, respectively; $I(i, j)$ represents a pixel in the binary image; i and j are the iterators for each row and column, respectively; and finally, let R and C be the total number of columns and rows of the image. Note that considering 8-bit images, the accumulated values are divided by $(255 \cdot [C, R])$ to normalize the results into the $[0-1]$ range. This avoids, at least partially, any limitation on the size of the used images, as the accumulated values do not depend on the width or height of the images.

The accumulated values can help to identify the differences between longitudinal and transverse cracks, as can be observed graphically in Figure 2, where the normalized accumulated values are represented with vertical and horizontal heatmap bars.

Similarly, the result of applying the accumulation Equations (1) and (2) can be observed in Figure 3 for an alligator crack sample and a healthy pavement one. Figure 3 shows graphically that the accumulated values are a powerful tool to differentiate between alligator cracks and healthy pavement since in this last case all cumulative values are very close to 0.

However, looking deeper into the results of the accumulations, as can be seen in Figures 2 and 3, there is still some information that is irrelevant for the classification because most of the values are 0, especially in healthy pavements, longitudinal cracks, and transverse cracks. Equations (1) and (2) reduce the feature space from $\{R \cdot C\}$ to $\{R + C\}$, but this new set of features still has a strong correlation with the Figure 3 example of accumulated pixels and the newly acquired features for alligator and healthy pavement classes location of the crack in the image and is not extensible to any arbitrary image resolution since the number of features of the output is always $\{R + C\}$.

Hence, in order to overcome these two undesirable limitations, a feature reduction space procedure is applied again. This time, the normalized accumulated values are the inputs of Equations (3) and (4). These two equations transform the feature space in a simple process to generate understandable features to classify the images into alligator cracks, transverse cracks, longitudinal cracks, and healthy pavement.

$$\max V = \max_{i=1}^{R}\left(Row_i^{Acc}\right) \qquad \max H = \max_{j=1}^{C}\left(Colm_j^{Acc}\right) \tag{3}$$

$$dV = \max V - \frac{\sum_{j=1}^{R} Row_j^{Acc}}{C}$$

$$dH = \max H - \frac{\sum_{i=1}^{C} Colm_i^{Acc}}{R} \tag{4}$$

Let $\max V$ and $\max H$ be the maximum values of the accumulated row and column values, respectively. They are computed by a function denoted by max. dV and dH are the difference between the maximum and the mean value of the $Row^{Acc}(I_j)$ and $Colm^{Acc}(I_i)$ respectively; i and j are the iterators to inspect the values of each vector $\forall i = 1, \ldots, C$; $\forall j = 1, \ldots, R$.

After applying the previously stated equations, each image has been reduced to only four features $\{\max V, dV, \max H, dH\}$, which describe the main characteristics of healthy pavements and the types of cracks analyzed in this work. Also, as Figures 2 and 3 show, these new four features are interpretable and explainable, which enables the possibility of creating simpler machine learning models than with other proposals. In contrast, using DDR approaches the complexity of the process depends on the size of the images and the relationship of the patterns in the whole dataset. In addition, feature reduction techniques do not provide interpretable datasets, which is an undesirable effect that can affect the performance of machine learning models and experts' understanding of how those models are using the data.

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6 | DATASETS

This section explains how the five datasets used in the experimentation were created. The first subsection shows the information from the datasets proposed by other authors in the field of application of this work, which are used later in this study. The second subsection shows the methodology followed for combining those datasets to create new datasets.

6.1 | Previous proposed datasets

This work pursues a DDR that allows, with interpretable features, the classification of cracks in pavements into four classes: transverse cracks, longitudinal cracks, alligator cracks, and healthy pavement. The datasets selected to carry out this study are publicly accessible:

Cubero-Fernandez et al. (2017): This dataset consists of healthy pavement images and cracks images labeled as alligator, longitudinal, and transverse. In total, this dataset has 600 labeled images with a resolution of 320 × 320 with 100 alligator crack images, 200 longitudinal crack images, 200 transverse crack images, and 100 healthy pavement images.

Rodriguez-Lozano et al. (2020): This dataset consists of 1846 images with a resolution of 320 × 320 containing 280 alligator cracks, 390 longitudinal cracks, 358 transverse cracks, and 828 images of healthy pavement.

AEL (Amhaz et al., 2016): the Aigle-RN & ESAR & LCMS (AEL) dataset consists of 58 images with different resolutions (991 × 462, 311 × 462, 768 × 512, 700 × 1000). However, the images are not labeled into classes as in the works of Cubero-Fernandez et al. (2017) and Rodriguez-Lozano et al. (2020). Hence, to use this dataset in this paper, we had to manually analyze and classify each image, obtaining 18 longitudinal cracks, 34 transverse cracks, and three healthy pavement images. Also, we included some new images in this dataset: five longitudinal cracks and three transverse cracks from TEMPEST2 and LRIS datasets, respectively. The remaining images cannot be classified into any of the four types described in Figure 1. Therefore, these images are not included in any of the datasets on this list.

Crack Forest (Y. Shi et al., 2016): This dataset contains 118 unlabeled images with a resolution of 480 × 320 pixels. We analyzed and classified manually the images, obtaining five alligator cracks, 10 longitudinal cracks, and 95 transverse cracks images.

Crack200 (Zou et al., 2012): This dataset contains 206 images, with a resolution of 800 × 600. From this dataset, we labeled 82 images obtaining 64 longitudinal crack images and 18 transverse crack images.

GAPS384 (Eisenbach et al., 2017; Yang et al., 2020): This dataset contains a modified portion of the dataset German Asphalt Pavement Distress. It contains 509 images with a resolution of 540 × 440 pixels. We identified 343 longitudinal cracks, 117 transverse cracks, and a healthy pavement image.

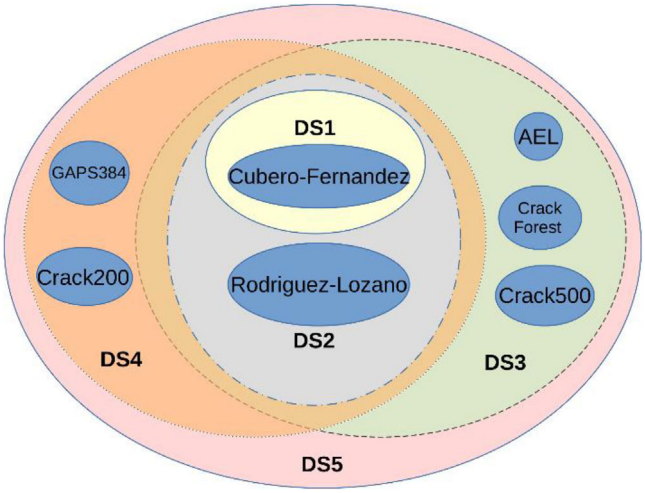

FIGURE 4 Datasets relations.

Crack500 (Yang et al., 2020; L. Zhang et al., 2016): This dataset contains 1896 images with a resolution of 360 × 640, where we labeled seven alligator cracks, 388 longitudinal cracks, 674 transverse cracks, and two healthy pavement images.

Finally, for the sake of reproducible research experimentation, we have decided to offer publicly all the labels of the images described above, which can be found in a public repository (Rodriguez-Lozano, 2023).

6.2 | Dataset creations

Some datasets are very small or do not cover the four types of cracks analyzed in this work. Therefore, the data used in this work are new datasets created from a combination of the original ones. The new dataset composition is created using an incremental approach as shown in Figure 4. It is detailed as follows:

1. Dataset 1 (DS1): This dataset contains images collected from the work of (Cubero-Fernandez et al., 2017). Initially, this dataset can be considered as the simplest case or as a "naïve" dataset, because the cracks that are included are mostly perfect with no inaccuracies, no partial combination of other cracks, or completely healthy pavement, without any imperfection in the surface.
2. Dataset 2 (DS2): This is the dataset used in the work of Rodriguez-Lozano et al. (2020). This dataset includes the previous dataset (DS1), along with some other images of different cracks and healthy pavement. With

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 1 Number of images in each dataset

	Alligator	Longitudinal	Transverse	Healthy
Dataset 1 (DS1)	100	200	200	100
Dataset 2 (DS2)	380	590	558	928
Dataset 3 (DS3)	392	1011	1364	933
Dataset 4 (DS4)	380	997	693	929
Dataset 5 (DS5)	392	1418	1499	934

this inclusion, data are not as perfect as in DS1 since the cracks of Rodriguez-Lozano et al. and the healthy pavement images have some small imperfections.

3. Dataset 3 (DS3): In this case, the previous datasets (DS1 and DS2) are extended with the labeled data from AEL (Amhaz et al., 2016), Crack Forest (Y. Shi et al., 2016), and Crack500 (Yang et al., 2020; L. Zhang et al., 2016) datasets. This addition adds more difficulty to the data and also generates an unbalanced dataset, where the alligator class is in the minority.
4. Dataset 4 (DS4): This dataset follows a similar organizational scheme as the previous dataset (DS3) taking the DS1 and DS2 as base datasets. However, in this case, the additions are GAPS384 (Eisenbach et al., 2017; Yang et al., 2020) and Crack200 (Zou et al., 2012) datasets. Similar to the case of DS3, this incorporation creates two minority classes, the alligator and the transverse classes.
5. Dataset 5 (DS5): In order to consider all the possible data, the last dataset is the combination of all the original datasets: Cubero-Fernandez et al. (2017), (Rodriguez-Lozano et al. (2020), AEL (Amhaz et al., 2016), Crack Forest (Y. Shi et al., 2016), Crack500 (Yang et al., 2020; L. Zhang et al., 2016), GAPS384 (Eisenbach et al., 2017; Yang et al., 2020), and Crack200 (Zou et al., 2012). Thus, DS5 is the most challenging dataset among all the five datasets considered in the experimental phase.

The number of images of different types in each of the generated datasets can be seen in Table 1.

7 | EXPERIMENTATION AND DISCUSSION

In this section, the results of the experiments are stated. The DDR4CC proposal is compared with the other DDR methods applied as a prior step to the classification stage.

TABLE 2 Number of features

	DS1	DS2	DS3	DS4	DS5
Linear discriminant analysis (LDA)			2		
Principal component analysis (PCA)	18	36	38	43	41
PCA-4			4		
Singular value decomposition (SVD)			4		
t-Distributed Stochastic Neighbor Embedding (TSNE)			3		
Multidimensional scaling (MDS)			4		
IVIS			4		
Principal projections (PP)			1120		
Data dimensionality reduction for crack classification algorithms (DDR4CC)			4		

Therefore, we will check how each DDR applied impacts each different classification method in terms of the quality of the classification with five incremental datasets. Afterward, we will provide the results of the timing taken by each DDR method and classification. Finally, we will make a joint comparison of both the quality of the classification and the timing.

7.1 | Classification performance analysis

To carry out a performance analysis of our proposal, we have used the five datasets generated from the previous section and employed the classification algorithms detailed in Section 4. In addition, in order to compare results, we have used the data reduction methods of Section 3 and the proposal of Cubero-Fernandez et al. (2017), Hoang (2018), and Rodriguez-Lozano et al. (2020) under the name "PP" on the same datasets.

However, to set a fair comparison of our proposal with the methods presented in Section 3 and PP, all images in each dataset are set to the same resolution: 640 × 480. Hence, after applying each data reduction algorithm, the total number of features is presented in Table 2.

It should be noted that the number of features in the above list was set for each of the methods to the number shown except for LDA, which in the most common cases uses two features to perform a linear separation of the data. In addition, there are two different PCA analyses: PCA where a number of features have been chosen to ensure

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

95% of the variability of the data, and PCA-4 where only 4 features have been taken out.

Once the features for each data reduction technique were obtained, 10-fold cross-validation was applied to each dataset to analyze the classification performance of the algorithms detailed in Section 4. In order to measure the performance, we selected the F1-score metric (Kulkarni et al., 2020), which is calculated following Equation (5).

$$F_1 - Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (5)$$

Let TP be the number of instances that the classifier has identified correctly; FP the number of instances that the classifier has identified as a specific class and which do not belong to that class; FN the number of instances that the classifier identified not belonging to a specific class, which were incorrect.

Table 3 shows the classification results for each dataset and machine learning algorithm, applying all DDR methods. This table shows that for DS1, which is the simplest, smallest dataset, and with balanced classes, the LDA method provides the most accurate results, compared to the other methods. Also, DDR4CC is slightly behind LDA for the F1-score metric, with a difference of 0.007, 0.007, 0.002, and 0.178 for Naïve Bayes, BFTree, PART, and SVM algorithms, respectively.

However, for the rest of the datasets, where the images are complex and classes are unbalanced, DDR4CC outperforms the results of most of the methods in all machine learning algorithms. It can be observed that there are some cases where the result provided by DDR4CC is not the highest one, but in the worst case, the difference is very small, with a value of 0.015 for DS2 using the Naïve Bayes algorithm. Additionally, it is important to note that PP is not able to work correctly with the Naïve Bayes algorithm because it is not possible to obtain an adequate value of the F1-score metric since there is, at least, one class for which the algorithm is not able to learn how to classify it with such input data.

Moreover, this table not only shows information about the classification performance of the proposed DDR approaches but also shows that there are algorithms such as SVM that in general provide less accurate results than the rest. Similarly, all the results show that the best-performing algorithms are usually PART and KNN as (Ahmadi et al., 2018) obtained.

7.2 | Timing performance analysis

The previous section analyzed the performance of our proposal, compared to others in terms of the quality of the classification. However, a comparison taking into account computing time spent should also be performed.

To carry out the timing analysis and given that one of the main objectives is its execution in systems with low computational capacity, we selected a low-resource embedded hardware "Nvidia Jetson Nano." This device is powered by an ARM A57 at 1.43 GHz and 4 GB LPDDR4. Also, this device is capable of executing machine learning and computer vision algorithms to perform the detection step, the feature extraction step, and the classification step, previously shown in Figure 1.

For the timing analysis, we have considered two different approaches:

1. The first approach is to check how much time each of the proposals takes as the image resolutions change. This allows visualizing which of those DDR methods scale worse as the image size increases.
2. The second approach is to analyze how long the machine learning algorithms take with each of the DDR across all datasets. This allows analyzing which of the algorithms in conjunction with each DDR method is the fastest, thus estimating an indication of the complexity of the models.

For the first approach, 50 random images from DS5 were used. After selecting the images, they were resized from a resolution of 64×48 and doubled up to an image resolution of 2560×1920. These resolutions cover a wide range of values to be able to analyze the behavior. For each resolution in that range, each data reduction method has been applied 100 times in order to avoid possible fluctuations in the measured times caused by the execution of any system task.

The results presented in Table 4 show that IVIS, MDS, and TSNE take much longer to perform the generation of all the data than the rest of the techniques. The behavior of the implemented experiments shows that the time taken by IVIS to process the images is independent of their resolution, taking on average about 33 s to process all the images. Next, we find SVD, another method that performs even worse than MDS and TSNE since when a resolution of 192×144 is reached, it takes more than 1 s to process all the images. In contrast to this, both PCA and LDA perform better, but unlike DDR4CC, both methods have a much steeper time increment, especially, at small resolutions. Furthermore, Table 4 shows that both PP and DDR4CC scale best as the image resolution increases. In fact, their behavior is practically linear because there are no abrupt changes, and it scales gradually as the resolution increases. Finally, Table 4 shows that the time differences between PP and DDR4CC are minimal, with DDR4CC taking an average of only 0.0003 seconds longer than PP.

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 3 Classification results

DS	Algorithm	LDA	PCA	PCA-4	SVD	TSNE	MDS	IVIS	PP	DDR4CC
1	Naïve Bayes	**1.000**	0.795	0.762	0.763	0.706	0.660	0.661	-	0.993
	K nearest neighbor (KNN)	**1.000**	0.983	0.938	0.937	0.940	0.853	0.905	**1.000**	**1.000**
	Best First Tree (BFTree)	**1.000**	0.957	0.940	0.942	0.832	0.833	0.780	0.916	0.993
	Partial decision trees (PART)	**0.995**	0.935	0.932	0.937	0.850	0.796	0.783	0.958	0.993
	Support vector machines (SVM)	**1.000**	0.556	0.533	0.532	0.705	0.599	-	0.743	0.822
2	Naïve Bayes	**0.951**	0.916	0.887	0.888	0.792	0.806	0.796	-	0.936
	KNN	0.942	0.892	0.951	0.951	0.984	0.848	0.948	0.984	**0.985**
	BFTree	0.943	0.946	0.938	0.938	0.933	0.827	0.919	0.933	**0.969**
	PART	0.944	0.954	0.933	0.932	0.943	0.826	0.929	0.955	**0.972**
	SVM	0.945	0.846	0.824	0.827	0.741	0.819	0.760	0.849	**0.953**
3	Naïve Bayes	0.882	0.870	0.782	0.783	0.879	0.672	0.773	-	**0.921**
	KNN	0.833	0.935	0.932	0.932	0.963	0.880	0.928	**0.964**	0.963
	BFTree	0.872	0.915	0.908	0.910	0.956	0.855	0.900	0.930	**0.958**
	PART	0.878	0.934	0.921	0.918	**0.960**	0.855	0.902	0.942	0.957
	SVM	0.853	0.614	0.571	0.570	0.867	0.576	0.693	0.674	**0.932**
4	Naïve Bayes	0.920	0.894	0.895	0.895	0.776	0.722	0.789	-	**0.946**
	KNN	0.887	0.889	0.934	0.935	0.972	0.817	0.907	0.974	**0.976**
	BFTree	0.905	0.936	0.928	0.928	0.939	0.802	0.871	0.925	**0.971**
	PART	0.910	0.946	0.930	0.933	0.937	0.800	0.871	0.944	**0.965**
	SVM	0.908	0.723	0.659	0.660	0.735	0.664	0.734	0.732	**0.940**
5	Naïve Bayes	0.843	0.880	0.775	0.778	0.899	0.624	0.852	-	**0.924**
	KNN	0.789	0.920	0.918	0.917	0.968	0.852	0.923	**0.963**	0.959
	BFTree	0.840	0.920	0.905	0.910	0.954	0.839	0.908	0.930	**0.954**
	PART	0.844	0.928	0.916	0.911	0.961	0.853	0.912	0.951	**0.961**
	SVM	0.792	0.643	0.624	0.624	0.905	0.630	0.688	0.655	**0.931**

Note: (-) represents a total failure classification in one of the classes. Best results in boldface.

TABLE 4 Evolution of time spent by DDR methods (in seconds)

Resolution	LDA	PCA	PCA-4	SVD	TSNE	MDS	IVIS	PP	DDR4CC
64 x 48	0.0866	0.0554	0.0547	0.0462	0.7766	0.4249	40.9766	0.0106	0.0108
320 x 240	0.3029	0.2684	2.2162	1.5622	0.8526	0.6919	40.8243	0.0399	0.0400
640 x 480	**0.4735**	**0.4158**	**2.5839**	**2.0255**	**0.9546**	**0.8722**	**37.7043**	**0.1200**	**0.1202**
1280 x 960	0.9431	0.8094	3.1503	2.4759	1.3396	1.3302	26.9136	0.4492	0.4495
1920 x 1440	1.8447	1.4014	3.3937	2.9038	1.9011	2.0290	31.7317	0.9726	0.9729
2560 x 1920	2.6500	2.1466	4.3242	3.7705	2.6749	2.8080	24.1659	1.6898	1.6902

For the second approach stated earlier, we measured the execution times taken by the algorithms described in Section 4 for each data reduction method. For this, models of each algorithm have been trained and used to classify 100 images. As in the previous case, each run was repeated 100 times to avoid the effect on the measured time of any running tasks in the system except for the classification experiment.

Table 5 shows a structure similar to Table 3. In this case, the contents are not the classification performance but the

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 5 Classification timing (milliseconds)

DS	Algorithm	LDA	PCA	PCA-4	SVD	TSNE	MDS	IVIS	PP	DDR4CC
1	Naïve Bayes	2.3520	9.7770	5.2130	3.2010	2.7490	2.6140	2.5200	351.6760	**1.1000**
	KNN	92.4130	39.3260	23.3370	18.3070	16.7840	16.9430	**16.7520**	221.6790	20.1000
	BFTree	0.3250	0.8040	0.3060	0.1900	0.2030	0.2350	0.1930	1.1930	**0.1410**
	PART	10.4900	3.9470	2.0680	0.9770	1.1000	1.2640	0.7970	1.0930	**0.2480**
	SVM	25.1750	28.7890	19.1220	28.1080	22.1150	10.5530	9.7380	157.5180	**6.1980**
2	Naïve Bayes	**0.5890**	6.8960	0.8090	0.7850	1.6480	2.4870	2.3060	337.6290	1.0300
	KNN	111.2290	193.4870	126.7390	108.1570	74.2840	58.9160	61.3680	966.0470	**51.1500**
	BFTree	0.2010	0.3770	0.1520	0.2410	0.1140	0.1200	**0.0870**	0.6540	0.0950
	PART	0.2510	0.2870	02960	0.3420	0.3760	0.3210	0.2920	0.8360	**0.2270**
	SVM	**8.7240**	170.4380	18.3150	42.0910	65.5060	19.2230	14.0080	788.4210	13.0110
3	Naïve Bayes	**0.6610**	8.5580	0.7640	0.9800	0.7050	0.8130	0.9330	262.8120	1.1140
	KNN	54.0570	61.3870	55.5880	58.5220	61.4950	54.8980	53.8630	807.6850	**50.6220**
	BFTree	**0.0710**	0.1390	0.0760	0.1280	0.1120	0.1130	0.1160	0.5940	0.0950
	PART	0.3070	0.4000	0.3230	0.4180	0.3510	0.3500	0.4780	1.2610	**0.2680**
	SVM	19.2440	253.3800	30.2410	27.0570	30.2750	27.8720	25.3180	502.7120	**19.0880**
4	Naïve Bayes	**0.5480**	10.6730	1.5980	1.2680	0.9880	0.9530	0.9910	261.8710	0.9000
	KNN	42.6970	48.2660	44.0540	50.0240	43.6080	43.0480	42.6930	484.1880	**42.5830**
	BFTree	**0.0710**	0.1480	0.1170	0.0850	0.1580	0.1120	0.1180	23.0580	0.0870
	PART	0.2920	2.2350	**0.2120**	0.2670	0.2810	0.3280	0.3010	0.7570	0.2410
	SVM	**12.8930**	233.0670	25.3710	26.1470	41.8870	30.6660	22.7440	228.2170	17.2770
5	Naïve Bayes	**0.5020**	8.0700	0.9280	1.1000	0.7100	1.2900	0.9130	255.7260	0.9170
	KNN	**58.5340**	69.9560	60.5650	62.7510	64.3780	60.6160	60.6620	900.0370	61.7290
	BFTree	**0.0780**	0.1400	0.0830	0.0970	0.1090	0.1090	0.1190	0.4850	0.0840
	PART	0.2450	0.5530	0.3590	0.2500	0.2330	0.3860	0.3220	1.1940	**0.2080**
	SVM	30.3080	283.1990	38.5110	43.9170	28.7470	39.8030	25.7420	230.0830	**19.9860**

Note: Fastest results in boldface.

execution times, in milliseconds, taken by each method and technique to classify 100 images.

As shown by the results in this table, LDA and DDR4CC offer the lowest times. As the results show, LDA provides in many cases slightly faster results than DDR4CC. This is mainly because LDA generates a new set of features in which there are only two features, unlike DDR4CC, which generates four. This might seem to be a disadvantage of DDR4CC; however, it should be remembered that in Table 3, DDR4CC outperformed all methods in most of the datasets, so if both tables are taken into consideration, DDR4CC clearly provides the best results overall. This is demonstrated in Section 7.3.

Table 5 shows that methods such as PCA provide worse time results by generating more complex models in which more calculations have to be performed to arrive at the classification of a new pattern. The same can be extrapolated to the PCA-4, SVD, TSNE, MDS, and IVIS methods since, despite containing three or four features, the generated models are much more complex due to the low interpretability of the data. This causes longer execution times. So, PP method generally obtains the worst results, taking in many cases 100 times longer than DDR4CC.

Table 5 shows for all datasets, both KNN and SVM are the most expensive methods to run. On the contrary, BFTree and PART are the most efficient methods to classify the data. Therefore, when choosing a method to carry out the classification, PART would be chosen because it is the one that provides the best results in terms of classification and invested time.

The DDR4CC method is specific for crack classification, it is not a generic option because it depends on the data interpretability. Thus, it is able to include expert knowledge in the selection of the most relevant features. Methods such as LDA, PCA, PCA-4, SVD, TSNE, MDS, and IVIS are mathematically robust, but they rely on the provided data. If the data are not complete or the given set of samples does not cover the real variability of the possible occurrences, the reduction will not be completely efficient. However, the DDR techniques specifically designed for crack classification include expert knowledge, and the selected features tend to be more efficient. For instance,

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

PP method, which is a specifically designed DDR method for crack classification, provides better results than the rest of the mathematical methods in most cases. However, DDR4CC outperforms PP method in most cases in terms of classification performance. In timing comparison, DDR4CC provides faster results than the PP method in any case.

7.3 | Joint discussion of the performance and timing

The aim of any classification is to provide the best results in terms of the performance of the quality of that classification. However, if the time taken to provide the results is too high, a reduction of the number of features required for the classification is applied. Usually, this reduction may provoke a degradation of the performance of the classification. Therefore, it is needed further analysis on both the time required for the full processing and the quality obtained.

After a preliminary analysis of the timing, the IVIS method obtains average classification results and has very large processing times. The processing times of the IVIS method along with any classification algorithm are much higher than any other DDR method. Therefore, we have decided to remove IVIS from the graphical comparative. In the following figures, we are comparing each classification technique with every DDR method. On the left vertical axis is shown the classification performance using F1-score, while on the right vertical axis is shown the timing in milliseconds. The timing comprises both the time taken by the DDR method applied on 640×480 images, as described in Table 4, and the classification time, stated in Table 5.

Figure 5 shows the graphical analysis of the Naïve Bayes classification technique with all the DDR methods under the DS5 dataset. In it, DDR4CC provides the best results in both the F1-score and the timing. Other techniques, such as TSNE and PCA, give good F1-Score results, although the timing is much worse than with DDR4CC.

We present in Figure 6 a comparison of the DDR methods for the classification algorithm KNN and DS5 dataset. The best classification results are obtained for TSNE, PP, and DDR4CC methods. DDR4CC outfits clearly the other methods. So, TSNE takes a total time of 1018.98 ms (DDR process takes 954.6 and 64.38 ms for classification); PP takes 1020.04 ms (120 ms for DDR and 900.04 ms for classification); and DDR4CC takes only 181.93 ms (120.2 ms for DDR and 61.73 for classification). We can conclude that applying KNN for a similar F1-score, DDR4CC is approximately 5.6 times faster than TSNE and PP.

Likewise, we present in Figure 7 the comparison of the DDR methods for the classification algorithm BFTree and

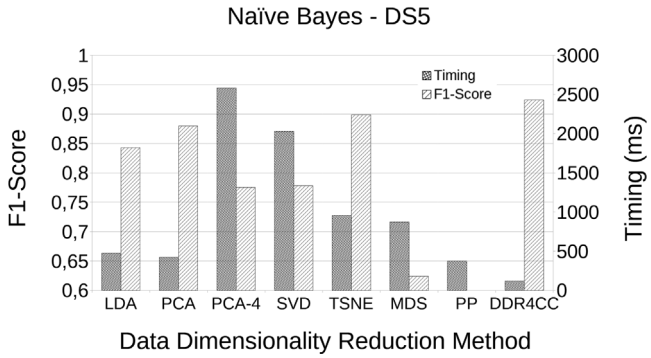

FIGURE 5 Comparison of data dimensionality reduction (DDR) techniques applied to the Naïve Bayes classifier regarding timing and F1-score performance in the Dataset 5 (DS5) dataset. DDR4CC, data dimensionality reduction for crack classification algorithms; LDA, linear discriminant analysis; PCA, principal component analysis; PP, principal projections; SVD, singular value decomposition; TSNE, t-distributed stochastic neighbor embedding.

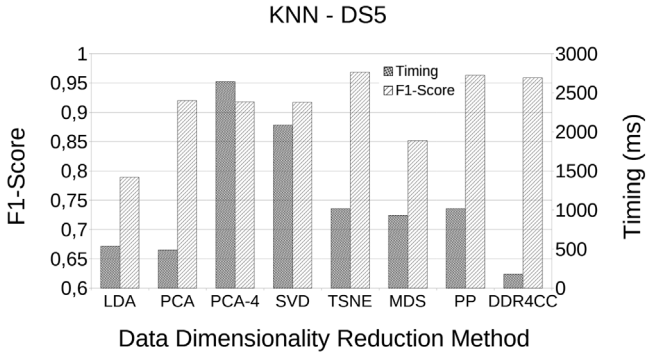

FIGURE 6 Comparison of DDR techniques applied to K nearest neighbor (KNN) classifier regarding timing and F1-score performance in the DS5 dataset. DDR4CC, data dimensionality reduction for crack classification algorithms; LDA, linear discriminant analysis; PCA, principal component analysis; PP, principal projections; SVD, singular value decomposition; TSNE, t-distributed stochastic neighbor embedding.

DS5 dataset. Best classification results are obtained for TSNE and DDR4CC methods. TSNE takes a total time of 954.7 ms (DDR process takes 954.6 and 0.109 ms for classification); DDR4CC takes only 120.3 ms (120.2 ms for DDR and 0.084 ms for classification). We can conclude that applying BFTree for a similar F1-score, DDR4CC is approximately 7.9 times faster than TSNE. DDR4CC again outperforms clearly the other methods.

The last comparison is presented in Figure 8, and this is the comparison of the DDR methods for the PART classification algorithm. The best classification results are obtained for TSNE, PP, and DDR4CC methods. TSNE takes a total time of 954.82 ms (DDR process takes 954.6 and 0.223 ms for classification); PP takes 121.94 ms (120 ms for DDR and 1.194 ms for classification); DDR4CC takes only 120.41

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

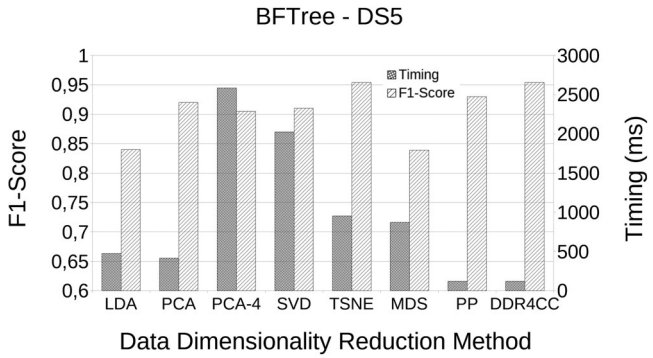

FIGURE 7 Comparison of DDR techniques applied to Best First Tree (BFTree) classifier regarding timing and F1-score performance in the DS5 dataset. DDR4CC, data dimensionality reduction for crack classification algorithms; LDA, linear discriminant analysis; PCA, principal component analysis; PP, principal projections; SVD, singular value decomposition; TSNE, t-distributed stochastic neighbor embedding.

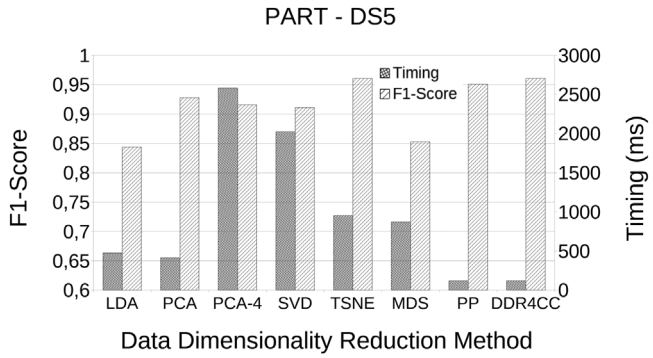

FIGURE 8 Comparison of DDR techniques applied to partial decision trees (PART) classifier regarding timing and F1-score performance in the DS5 dataset. DDR4CC, data dimensionality reduction for crack classification algorithms; LDA, linear discriminant analysis; PCA, principal component analysis; PP, principal projections; SVD, singular value decomposition; TSNE, t-distributed stochastic neighbor embedding.

ms (120.2 ms for DDR and 0.208 ms for classification). Even though PP is very close this time, DDR4CC again obtains the best results.

Similar results than those of Naïve Bayes classifier (shown in Figure 5) are obtained for the SVM classification algorithm, as shown in Figure 9, where again DDR4CC gets the best results, while all the other methods obtain lower F1-score and higher execution time. In addition, the 225 confusion matrices resulting from the execution of all experiments are available in the file "ConfusionMatrixResults.txt" in the "DS" folder of the public repository (Rodriguez-Lozano, 2023).

We can assert after analyzing these five classification algorithms, and testing for nine DDR methods, that the DDR4CC data reduction technique obtains the best results

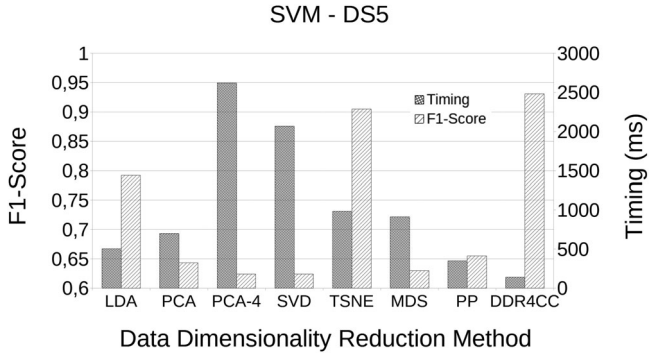

FIGURE 9 Comparison of DDR techniques applied to support vector machines (SVM) classifier regarding timing and F1-score performance in the DS5 dataset. DDR4CC, data dimensionality reduction for crack classification algorithms; LDA, linear discriminant analysis; PCA, principal component analysis; PP, principal projections; SVD, singular value decomposition; TSNE, t-distributed stochastic neighbor embedding.

in execution time and provide the best input data for those classification algorithms, obtaining the highest F1-score.

8 | CONCLUSION

The number of features used for pavement crack classification algorithms is usually high, poorly interpretable, and difficult to obtain. As these features are the ones used by machine learning algorithms to learn from the data, they generate more complex models. Therefore, they are highly penalized by the classification accuracy and the time spent in classifying the data.

This work proposes the DDR4CC method to reduce the dimensionality of the data as a step before the classification. DDR4CC is a simple DDR method with high interpretability of the data. Thanks to this, we obtain very fast computation and high accuracy.

We demonstrate in Section 7.3 that the DDR4CC data reduction technique obtains the best results in execution time and provide the best input data for those classification algorithms, obtaining the higher F1-score. The 225 confusion matrices obtained are available in (Rodriguez-Lozano, 2023).

DDR4CC avoids using thousands of features to identify longitudinal, transverse, alligator cracks, and healthy pavement as opposed to the behavior of other methods. Specifically, DDR4CC generates only four features: $\{\max V, dV, \max H, dH\}$. This set of features has multiple advantages. One advantage is that these features are not influenced by the spatial location of the defect in the images. In addition, they allow the extraction of such features regardless of the original image resolution.

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Finally, these features boost the classification algorithms and improve the execution time to classify the image.

So, two main benefits are obtained: Better classification is reached in the majority of the experiments. The reduction of computation will allow real-time detection and classification on low-consumption devices. This could really be useful in unmanned aerial vehicules (UAVs).

To analyze the DDR4CC performance, several comparisons have been made with five datasets created by blending different state-of-the-art public datasets. Many of these datasets did not classify images into longitudinal cracks, transverse cracks, alligator cracks, or healthy pavement. Therefore, in conjunction with this work, open public access is provided to the classification index files for each of the images identified in one of the abovementioned types.

In addition, eight different DDR techniques have been applied to each of the datasets to test their effectiveness against DDR4CC. For this purpose, the performance of DDR4CC has been analyzed from a classification performance point of view using five machine learning algorithms (some as simple as Naïve Bayes). From the obtained results, it can be deduced that, in general, DDR4CC boosts the machine learning algorithms obtaining almost perfect models and better classification results according to the F1-score metric.

On the other hand, the temporal impact of DDR4CC related to the rest of the DDR algorithms and of the classification algorithms on low computational capacity embedded systems has been analyzed. In the comparison of the DDR methods, it can be observed that DDR4CC has an almost linear behavior, scaling more uniformly and progressively as the size of the images increases than the rest of the compared methods. Finally, the time spent by the algorithms shows that DDR4CC, along with the LDA method, provides the best results in the execution time spent in the inference stage in the machine learning algorithms. However, taking into account both the classification ranking and timing performances, we can conclude that DDR4CC is the most efficient method in low computational capacity systems for the classification of cracks.

This work should be continued in future works. The first one focused on reaching real-time detection and classification of low-consumption devices installed in UAVs; we are starting an industry collaboration titled INSPECT-ROADS with SANDO in which we will apply our algorithms to street and road maintenance in Malaga County (Spain). The second one is to analyze the impact of DDR4CC in newer supervised machine learning or classification algorithms, such as the dynamic ensemble learning algorithm (Alam et al., 2020), and the finite element machine for fast learning (Pereira et al., 2020).

ACKNOWLEDGMENTS

This research is partially supported by the Vision Assisted Navigation for Autonomous Vehicles project—VISNAV (RFFINNL-298890) Norwegian Research Council, and by the Advanced Informatics Research Group – GIIA (TIC-252), Universidad de Córdoba (Spain).

Open Access Funding provided by Universidad de Córdoba / CBUA.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflicts of interest.

REFERENCES

Abdel-Qader, I., Pashaie-Rad, S., Abudayyeh, O., & Yehia, S. (2006). PCA-based algorithm for unsupervised bridge crack detection. *Advances in Engineering Software*, *37*(12), 771–778. https://doi.org/10.1016/j.advengsoft.2006.06.002

Adeli, H., & Wu, M. (1998). Regularization neural network for construction cost estimation. *Journal of Construction Engineering and Management*, *124*(1), 18–24. https://doi.org/10.1061/(ASCE)0733-9364(1998)124:1(18)

Ahmadi, A., Khalesi, S., & Bagheri, M. (2018). Automatic road crack detection and classification using image processing techniques, machine learning and integrated models in urban areas: A novel image binarization technique. *Journal of Industrial and Systems Engineering*, *11*(Special issue 1), 85–97.

Ahmadi, A., Khalesi, S., & Golroo, A. (2021). An integrated machine learning model for automatic road crack detection and classification in urban areas. *International Journal of Pavement Engineering*, *23*(10), 3536–3552. https://doi.org/10.1080/10298436.2021.1905808

Alam, K. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing with Applications*, *32*(10), 8675–8690. https://doi.org/10.1007/s00521-019-04359-7

Alfeilat, H. A. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Salman, H. S. E., & Prasath, V. S. (2019). Effects of distance measure choice on k-nearest neighbor classifier performance: A review. *Big RData*, *7*(4), 221–248. https://doi.org/10.1089/big.2018.0175

Ali, R., Chuah, J. H., Talip, M. S. A., Mokhtar, N., & Shoaib, M. A. (2022). Structural crack detection using deep convolutional neural networks. *Automation in Construction*, *133*, 103989. https://doi.org/10.1016/j.autcon.2021.103989

Amhaz, R., Chambon, S., Idier, J., & Baltazart, V. (2016). Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, *17*(10), 2718–2729. https://doi.org/10.1109/TITS.2015.2477675

Çelik, F., & König, M. (2022). A sigmoid-optimized encoder–decoder network for crack segmentation with copy-edit-paste transfer learning. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1875–1890.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–27. https://doi.org/10.1145/1961189.1961199

Chen, C., Seo, H., Jun, C. H., & Zhao, Y. (2021). Pavement crack detection and classification based on fusion feature of LBP and PCA

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

with SVM. *International Journal of Pavement Engineering*, *23*(9), 3274–3283. https://doi.org/10.1080/10298436.2021.1888092

Chen, J., & He, Y. (2022). A novel u-shaped encoder–decoder network with attention mechanism for detection and evaluation of road cracks at pixel level. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1721–1736.

Chu, H., Wang, W., & Deng, L. (2022). Tiny-crack-net: A multiscale feature fusion network with attention mechanisms for segmentation of tiny cracks. *Computer-Aided Civil and Infrastructure Engineering*, *37*(14), 1914–1931.

Chun, P.-J., Izumi, S., & Yamane, T. (2021). Automatic detection method of cracks from concrete surface imagery using two-step light gradient boosting machine. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 61–72.

Cohen, W. W. (1995). Fast effective rule induction. In A. Prieditis & S. Russell (Eds.), *Machine learning proceedings 1995* (pp. 115–123). Morgan Kaufmann. https://doi.org/10.1016/B978-1-55860-377-6.50023-2

Cubero-Fernandez, A., Rodriguez-Lozano, F. J., Villatoro, R., Olivares, J., & Palomares, J. M. (2017). Efficient pavement crack detection and classification. *EURASIP Journal on Image and Video Processing*, *2017*(1), 39. https://doi.org/10.1186/s13640-017-0187-0

Dhumal, P., & Deshmukh, S. S. (2016). Survey on comparative analysis of various image compression algorithms with singular value decomposition. *International Journal of Computer Applications*, *133*, 18–21.

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., & Gross, H.-M. (2017). How to get pavement distress detection ready for deep learning? A systematic approach. International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, (pp. 2039–2047).

Espindola, A. C., Rahman, M., Mathavan, S., & Júnior, E. F. N. (2022). Comparing different deep learning architectures as vision-based multi-label classifiers for identification of multiple distresses on asphalt pavement. *Transportation Research Record*. https://doi.org/10.1177/03611981221127273

Garber, N., & Hoel, L. (2008). *Traffic & highway engineering*. Cengage Learning.

Gonzalez, R. C., & Woods, R. E. (2018). Projections and the radon transformation. In V. Tiwari (Ed.), *Digital image processing* (4th ed., pp. 374–377). Pearson, New York, NY, USA.

Gou, J., Sun, L., Du, L., Ma, H., Xiong, T., Ou, W., & Zhan, Y. (2022). A representation coefficient- based k-nearest centroid neighbor classifier. *Expert Systems with Applications*, *194*, 116529. https://doi.org/10.1016/j.eswa.2022.116529

Hamishebahar, Y., Guan, H., So, S., & Jo, J. (2022). A comprehensive review of deep learning-based crack detection approaches. *Applied Sciences*, *12*(3), 1374. https://doi.org/10.3390/app12031374

Hoang, N.-D. (2018). Classification of asphalt pavement cracks using Laplacian pyramid-based image processing and a hybrid computational approach. *Computational Intelligence and Neuroscience*, *2018*, 1312787. https://doi.org/10.1155/2018/1312787

Hoang, N.-D., & Nguyen, Q.-L. (2018). A novel method for asphalt pavement crack classification based on image processing and machine learning. *Engineering with Computers*, *35*(2), 487–498. https://doi.org/10.1007/s00366-018-0611-9

Hou, Y., Dong, Y., Zhang, Y., Zhou, Z., Tong, X., Wu, Q., & Li, R. (2022). The application of a pavement distress detection method based on FS-Net. *Sustainability*, *14*(5), 2715. https://doi.org/10.3390/su14052715

Hout, M. C., Papesh, M. H., & Goldinger, S. D. (2013). Multidimensional scaling. *WIREs Cognitive Science*, *4*(1), 93–103.

Ibarguren, I., Lasarguren, A., Pérez, J. M., Muguerza, J., Gurrutxaga, I., & Arbelaitz, O. (2016). BFPART: Best-first part. *Information Sciences*, *367–368*, 927–952. https://doi.org/10.1016/j.ins.2016.07.023

Ibrahim, A., Osman, M., Yusof, N., Ahmad, K., Harun, N., & Raof, R. (2019). Characterization of cracking in pavement distress using image processing techniques and k-nearest neighbour. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(2), 810. 10.11591/ijeecs.v14.i2.pp810-818

Jana, S., Thangam, S., Kishore, A., Sai Kumar, V., & Vandana, S. (2022). Transfer learning based deep convolutional neural network model for pavement crack detection from images. *International Journal of Nonlinear Analysis and Applications*, *13*(1), 1209–1223. 10.22075/ijnaa.2021.24521.2762

Jang, K., An, Y.-K., Kim, B., & Cho, S. (2021). Automated crack evaluation of a high-rise bridge pier using a ring-type climbing robot. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 14–29.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Kong, S.-Y., Fan, J.-S., Liu, Y.-F., Wei, X.-C., & Ma, X.-W. (2021). Automated crack assessment and quantitative growth monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *36*(5), 656–674.

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). 5- foundations of data imbalance and solutions for a data democracy. In F. A. Batarseh & R. Yang (Eds.), *Data democracy* (pp. 83–106). Academic Press. https://doi.org/10.1016/B978-0-12-818366-3.00005-8

Kusumaningrum, J., Madenda, S., Karmilasari, & Nahdalina (2022). Detection and classification of road damage based on image morphology and k-NN method (k nearest neighbour). *International Journal of Engineering and Advanced Technology*, *11*(5), 86–90. https://doi.org/10.35940/ijeat.e3543.0611522

Li, W., Huyan, J., Gao, R., Hao, X., Hu, Y., & Zhang, Y. (2021). Unsupervised deep learning for road crack classification by fusing convolutional neural network and k_means clustering. *Journal of Transportation Engineering, Part B: Pavements*, *147*(4), 04021066. https://doi.org/10.1061/JPEODX.0000322

Liang, S., Jianchun, X., & Xun, Z. (2018). An extraction and classification algorithm for concrete cracks based on machine vision. *IEEE Access*, *6*, 45051–45061. https://doi.org/10.1109/ACCESS.2018.2856806

Liu, C., & Xu, B. (2022). A night pavement crack detection method based on image-to-image translation. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1737–1753.

Mai, Q. (2013). A review of discriminant analysis in high dimensions. *WIREs Computational Statistics*, *5*(3), 190–197.

Meng, S., Gao, Z., Zhou, Y., He, B., & Djerrad, A. (2022). Real-time automatic crack detection method based on drone. *Computer-Aided Civil and Infrastructure Engineering*, *38*(7), 849–872. https://doi.org/10.1111/mice.12918

Mokhtari, S., Wu, L., & Yun, H.-B. (2017). Statistical selection and interpretation of imagery features for computer vision-based pavement crack-detection systems. *Journal of Performance of*

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Constructed Facilities, *31*(5), 04017054. https://doi.org/10.1061/(asce)cf.1943-5509.0001006

Nanga, S., Bawah, A., Acquaye, B., Billa, M., Baeta, F., Odai, N., & Nsiah, A. (2021). Review of dimension reduction methods. *Journal of Data Analysis and Information Processing*, *9*, 189–231. https://doi.org/10.4236/jdaip.2021.93013

Osman, M. K., Zamree, M. E. A. M., Idris, M., Ahmad, K. A., Yusof, N. A. M., Ibrahim, A., & Bahri, I. (2021). Pavement crack classification using deep convolutional neural network. *Journal of Mechanical Engineering*, SI *10*(1), 227–244.

Palomar, R., Palomares, J. M., Castillo, J. M., Olivares, J., & Gomez-Luna, J. (2010). Parallelizing and optimizing lip-canny using nvidia cuda. In N. García-Pedrajas, F. Herrera, C. Fyfe, J. M. Benítez, M. Ali (Eds.), *Trends in applied intelligent systems* (pp. 389–398). Springer Berlin Heidelberg.

Pereira, D., Piteri, M., Souza, A., Papa, J., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. *Neural Computing with Applications*, *32*(10), 6393–6404. https://doi.org/10.1007/s00521-019-04146-4

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc.

Rafiei, M., & Adeli, H. (2017a). NEEWS: A novel earthquake early warning system using neural dynamic classification and neural dynamic optimization model. *Soil Dynamics and Earthquake Engineering*, *100*, 417–427.

Rafiei, M., & Adeli, H. (2017b). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083. https://doi.org/10.1109/TNNLS.2017.2682102

Rafiei, M., & Adeli, H. (2017c). A novel machine learning based algorithm to detect damage in highrise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400. https://doi.org/10.1002/tal.1400

Rodriguez-Lozano, F. J. (2023). Labeled-crack-dataset. https://github.com/FJ-Rodriguez-Lozano/Labeled-Crack-Dataset

Rodriguez-Lozano, F. J., Leon-Garcia, F., Gamez-Granados, J. C., Palomares, J. M., & Olivares, J. (2020). Benefits of ensemble models in road pavement cracking classification. *Computer-Aided Civil and Infrastructure Engineering*, *35*(11), 1194–1208. https://doi.org/10.1111/mice.12543

Sheerin, S. N., Kavitha, S., & Raghuraman, G. (2018). Review and analysis of crack detection and classification techniques based on crack types. *International Journal of Applied Engineering Research*, *13*(8), 6056–6062.

Shi, H. (2007). Best-first decision tree learning. [Unpublished master's thesis]. University of Waikato, Hamilton, NZ. COMP594.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, *17*(12), 3434–3445.

Szubert, B., Cole, J., Monaco, C., & Drozdov, I. (2019). Structure-preserving visualisation of high dimensional single-cell datasets. *Scientific Reports*, *9*, 8914. https://doi.org/10.1038/s41598-019-45301-0

Tsai, Y.-C., Zhao, Y., Pop-Stefanov, B., & Chatterjee, A. (2020). Automatically detect and classify asphalt pavement raveling severity using 3D technology and machine learning. *International Journal of Pavement Research and Technology*, *14*, 487–495. https://doi.org/10.1007/s42947-020-0138-5

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

Wan, F., Sun, C., He, H., Lei, G., Xu, L., & Xiao, T. (2022). YOLO-LRDD: A lightweight method for road damage detection based on improved yolov5s. *EURASIP J. Adv. Signal Process*, *2022*(1), 98. https://doi.org/10.1186/s13634-022-00931-x

Xie, X., Cai, J., Wang, H., Wang, Q., Xu, J., Zhou, Y., & Zhou, B. (2022). Sparse-sensing and superpixel-based segmentation model for concrete cracks. *Computer-Aided Civil and Infrastructure Engineering*, *37*(13), 1769–1784.

Xu, Z., Che, Y., Min, H., Wang, Z., & Zhao, X. (2018). Initial classification algorithm for pavement distress images using features fusion. In G. De Pietro, L. Gallo, R. J. Howlett, L. C. Jain, L. Vlacic (Eds.), *Intelligent interactive multimedia systems and services* (pp. 418–427). Springer International Publishing. https://doi.org/10.1007/978-3-319-92231-7_43

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2020). Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, *21*(4), 1525–1535. https://doi.org/10.1109/TITS.2019.2910595

Zhang, H., & Jiang, L. (2022). Fine tuning attribute weighted naive bayes. *Neurocomputing*, *488*, 402–411. https://doi.org/10.1016/j.neucom.2022.03.020

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ (pp. 3708–3712).

Zhang, Y., & Yuen, K.-V. (2021). Crack detection using fusion features-based broad learning system and image processing. *Computer-Aided Civil and Infrastructure Engineering*, *36*(12), 1568–1584.

Zheng, Y., Gao, Y., Lu, S., & Mosalam, K. M. (2022). Multistage semisupervised active learning framework for crack identification, segmentation, and measurement of bridges. *Computer-Aided Civil and Infrastructure Engineering*, *37*(9), 1089–1108.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). Crack-tree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238.

How to cite this article: Rodriguez-Lozano, F. J., Gámez-Granados, J. C., Palomares, J. M., & Olivares, J. (2023). Efficient data dimensionality reduction method for improving road crack classification algorithms. *Computer-Aided Civil and Infrastructure Engineering*, *38,* 2339–2354. https://doi.org/10.1111/mice.13014

14678667, 2023, 16, Downloaded from https://onlinelibrary.wiley.com/doi/10.1111/mice.13014 by Cbua-Consorcio De Bibliotecas, Wiley Online Library on [23/05/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License