Received ** **** ****; | Revised ** **** ****; | Accepted ** **** ****

DOI: xxx/xxxx

ORIGINAL ARTICLE

Benefits of ensemble models in road pavement cracking classification

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

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

***Correspondence**
Francisco J. Rodriguez-Lozano, Department of Electronic and Computer Engineering, Universidad de Córdoba, Edificio Leonardo da Vinci, Campus de Rabanales, 14071, Córdoba, Spain.
Email: fj.rodriguez@uco.es

Abstract

The maintenance of road pavements is an essential task to prevent major deterioration and to reduce accident rates. In this task, the detection and classification of different types of cracks on the roads is usually considered. However, in most cases these tasks are not fully automated and they need to be supervised by an expert in order to make repair decisions. This work focuses on the automatic classification of the most common types of cracks: longitudinal cracks, transverse cracks, and alligator cracks. Our proposal combines, firstly, computer vision techniques for crack segmentation and secondly, an ensemble model (composed of different rule–based algorithms) for the classification. This approach achieves an average precision and recall values greater than 94% for three analyzed datasets improving the results in comparison to other approaches.

KEYWORDS:
Image processing, Ensemble models, Road maintenance, Crack detection, Crack classification

1 | INTRODUCTION

Road pavement condition plays a fundamental role in the safety of vehicles (and in the future, with the self–driving vehicles (Foss, 2017)). In fact, this aspect has key implications in the acceleration and stability of vehicles (Bella, Calvi, & D'Amico, 2012). One of the key factors that directly affects pavement condition, is climate conditions (Schweikert, Chinowsky, Espinet, & Tarbert, 2014). Low temperatures (Chai, van Staden, Guan, Kelly, & Chowdhury, 2016; Galbraith RM, 2005) favor the appearance of alligator cracks, transverse cracks, longitudinal cracks, potholes, etc. In addition, another factor is the increase in traffic density (Mills, Tighe, Andrey, Smith, & Huen, 2009). Notice that in 2016, the number of motor vehicles throughout the world was greater than 2.1 billion (The World Health Organization, 2018). Hence, the maintenance of pavements could be an effective solution to solve the deterioration problem. Indeed, roads with adequate maintenance show a lower frequency of accidents (Lee, Nam, & Abdel-Aty, 2015).

Nevertheless, road pavement maintenance is expensive (Babashamsi, Yusoff, Ceylan, Nor, & Jenatabadi, 2016; Qiao, Chen, Alinizzi, & Labi, 2018) in economic terms and the tools for inspection and maintenance decision–making tasks are still far from automated. Recent studies (Radopoulou & Brilakis, 2017) show that the data acquisition of road maintenance can be performed automatically but the task of detection and classification of the defects remains a manual task by an expert in 99.6% of the cases, making this a time–consuming process.

In this context, computer vision (Rafiei & Adeli, 2017; Spencer, Hoskere, & Narazaki, 2019) provides useful methods that can be applied to road maintenance. These methods allow an image to be taken and then processed. For example, some techniques involve image enhancement (Long, 2008), illumination modification (Song & Cosman, 2018), edge detection (James, 2016), background subtraction (Nayyeri, Hou, Zhou, & Guan, 2018). These techniques can provide

a new data output from images that can feed machine learning algorithms. For example, algorithms such as unsupervised tools (Gutierrez Soto & Adeli, 2017; Rafiei & Adeli, 2018b), prediction tools (Rafiei & Adeli, 2018a; Rafiei, Khushefati, Demirboga, & Adeli, 2017), classification tools (Ibrahim et al., 2019; Z. Li, Cheng, Kwan, Tong, & Tian, 2019) or rule–based tools (Soto-Hidalgo, Alonso, Acampora, & Alcala-Fdez, 2018) that traditionally were carried out only by experts.

In the wide range of machine learning algorithm types, ensemble models (Zhou, 2012) take advantage of the use of several models so improving the classification task compared to the use of separate machine learning models. These models have not been widely exploited previously in pavement road crack classification. Also, ensemble models have been successfully applied to other fields in civil engineering problems (Prayogo, Cheng, Wu, & Tran, 2019; Sun, Li, & Adeli, 2013).

Hence, the main contribution of this paper is the application of an ensemble method composed of different rule–based algorithms and decision trees to improve current proposals in road pavement crack classification. The ensemble method carries out the classification of cracks into their most common types: transverse cracks, longitudinal cracks, and alligator cracks (Garber & Hoel, 2008). In order to provide the inputs to the proposed ensemble model, computer vision algorithms to extract the main features (without the necessity of previous training steps) are applied.

This paper is organized as follows: Section 2 shows different proposals focused on crack detection and classification. Section 3 describes the computer vision algorithms and the ensemble model (along with the algorithms that compose the ensemble) used to extract features of the cracks and their subsequent classification. The results and comparison with other authors are analyzed and discussed in Section 4. Finally, the main conclusions and future research are presented in Section 5 and Section 6 respectively.

2 | STATE OF THE ART

The crack detection and classification problem has taken on great importance, as the survey of Ragnoli, Blasiis, & Benedetto (2018) highlights. We have divided and structured different proposals into three different groups depending on their target. The first group is *'Road crack segmentation'*(subsection 2.1) where all the proposals are focused on the segmentation of the different types of cracks. The second group *'Single machine learning algorithm for road crack classification'*(subsection 2.2) is focused on the use of computer vision and single machine learning to classify the different types of cracks. The last group *'Multiple machine learning algorithm models for road crack classification'*(subsection 2.3) shares the same target as the previous group, although, these proposals use more than one machine learning algorithm to perform the classification task.

2.1 | Road crack segmentation

Crack segmentation has been researched using image processing along with machine learning approaches to identify the region to obtain a mask image (or binary image) where the cracks are defined by positive values. An example of this is proposed by T. Wang, Gopalakrishnan, Smadi, & Somani (2018). Their work is focused on the detection of longitudinal and transverse cracks based on the Potential Crack Components in an image (PCrCs). This analysis is carried out by filtering the intensity values between the pavement and the crack, deleting any non–connected components. Once the small elements have been eliminated, a crack is defined by a large connected area. To identify the crack and the background, the Shape Metric (SM), defined by the authors, is used. Compared to other methods such as Support Vector Machines (SVM), the authors obtain improved accuracy and a lower false positive rate.

Deep learning approaches have also been used in the crack detection problem to identify where a crack appears in an image, as in the proposal of Cha, Choi, & Büyüköztürk (2017). Their deep learning method is based on a Convolutional Neural Network architecture (CNN) with eight layers covering different steps from convolution, pooling, linear rectification, and softmax. However, their work is focused on the segmentation of concrete pavement cracks and does not extract the crack itself but generates a region of interest that includes the crack.

An alternative solution based on deep learning techniques is developed by Bang, Park, Kim, san Yoon, & Kim (2018) using a deep residual network with transfer learning. Their model is composed of 152 layers and performs a pixel–based classification of cracks and non–cracks. A similar approach is carried out by Cheng, Xiong, Chen, Gu, & Li (2018). The authors perform another pixel–based identification using a special type of CNN called U–NET. U–NET (Ronneberger, P.Fischer, & Brox, 2015) is defined as a type of convolutional net with fully connected layers, where part of the net is used as an encoder and the rest, as a decoder. Their work is focused on binary classification of crack or non–crack of each pixel in the image.

Notwithstanding, pixel–based classification is not exclusive to deep learning approaches. Ai, Jiang, Kei, & Li (2018) carry out this task using image processing in combination with supporting vector machines. The authors focus on determining if a pixel of an image is a component of a crack or non–crack. They use a data fusion method with probability maps based on

pixel intensity, and probability maps based on neighborhood information of each pixel.

A different approach of crack detection is presented by B. Wang et al. (2019) where the Extreme Learning Machine (ELM) model is applied. The advantage of this approach is that the parameters of the hidden layers are initialized and randomly selected. This makes ELM more flexible without requiring any expert knowledge. It takes less processing time in the training stages than conventional neural network models.

As the previously described proposals show, 2D images have been used in the crack segmentation task. Nevertheless, other researchers have utilized the advantages of data fusion. For example, Xu, Chen, Wu, & Li (2018) use a mixture of 2D images together with the use of four linear lasers to detect the cracks. Their proposal provides, in some cases, errors smaller than 0.78 mm in the recovery of the cracks. As the authors conclude, their work provides "application potentials in the vision–based pavement inspection".

2.2 | Single machine learning algorithm for road crack classification

The road crack classification problem is an important factor which, if a crack is identified in time, can help reduce the cost of pavement repairs (Babashamsi et al., 2016). This problem has been analyzed using a single machine learning model from two classes (Kim, Ahn, Shin, & Sim, 2018) to multiple classes.

An example of binary classification is proposed by Kim et al. (2018). This proposal comprises two stages: the calculation of Concrete Crack Regions (CCR) and the classification of binary images obtained from the CCR. The calculation of the CCR is based on a binarization of the image considering that the dark areas may be candidates for cracks, and the lighter ones, the background. A comparison between the SURF–based (Speeded Up Robust Features) classification and CNN is carried out.

Several proposals are focused on the analysis and comparison of different machine learning algorithms. For instance, Hoang & Nguyen (2018) analyze the cracks in pavements using steerable filters based on Gaussian filters and the second derivative of the image, obtaining binary images. The projections of binary images are calculated to reduce the data employed in the image classification. To detect alligator, longitudinal, and vertical cracks, the authors compare three different classification algorithms, Artificial Neural Networks (ANN), SVM, and Random Forest (RF). SVM models provide the most accurate results.

Another work that deals with the analysis of different classification algorithms is the proposal of Z. Li et al. (2019), where Low altitude Unmanned Aerial Vehicle Light Detection And Ranging systems (UAV LiDAR) are used. The authors obtain point clouds that are treated and applied as inputs for different classification algorithms such as SVM, RF, and Maximum Likelihood Classification (MLC). The authors analyze the damage to the pavement by classifying the deterioration of the pavement into cracks, potholes, subsidence, or rutting problems. The results show that the use of RF is the best option.

Deep learning approaches have been researched as the work of B. Li, Wang, Zhang, Yang, & Wang (2018) shows. Their research uses Convolutional Neural Networks (CNN) to detect longitudinal, transverse, block, and alligator type cracks. The authors use 3D images obtained by the PaveVision 3D system (Luo, Wang, Li, Li, & Moravec, 2014). The raw images are divided into smaller areas with a size of 512×512 pixels and are used directly as Neural Network inputs.

Fuzzy classification algorithms are also analyzed by Ibrahim et al. (2019). The authors perform a detection and classification of transverse and longitudinal cracks based on image processing algorithms and machine learning algorithms. Their proposal consists of a four–step process: image acquisition, image enhancement, feature extraction, and image classification. The authors use the K–Nearest Neighbors algorithm (KNN) in the classification step and a variant of this called fuzzy–KNN (Shang et al., 2005). Nevertheless, the image processing step is based on a threshold that must be fixed manually.

2.3 | Multiple machine learning algorithm models for road crack classification

Few proposals based on the use of multiple models can be found in the literature, although promising results can be seen in crack classification.

For instance, Cubero-Fernandez, Rodriguez-Lozano, Villatoro, Olivares, & Palomares (2017) present a system to detect and classify the longitudinal, transverse, and alligator cracks. Their work is based on edge detection using computer vision for processing crack images. After the processing, an induction tree model based on C4.5 is used. A C4.5 model is created to classify transverse cracks or non–cracks, and another C4.5 model to classify longitudinal cracks or non–cracks. Alligator cracks are classified using the information from these two models when longitudinal and transverse cracks are detected simultaneously.

Another approach for crack classification is proposed by L. Li, Sun, Ning, & Tan (2014). In the detection task, computer vision algorithms are used to correct the illumination obtaining a binary image of the crack. This image is used in the classification task to label each crack as linear or alligator type. Once the linear cracks are detected, they are separated into longitudinal and transverse cracks. A Back–propagation

Neural Network (BNN) algorithm and a decision rule are used in the classification stages.

The previous proposals in the field of road pavement cracks are focused on the use of deep learning methods for the segmentation task. In the classification task, the majority of the proposals use a single classification method and only a few of them use multiple classification models. However, to the best of our knowledge, ensemble methods have not been used previously in the pavement crack classification task. Hence, the following sections detail a methodology to extract the features of images needed as inputs to an ensemble method composed of rule–based algorithms and decision trees.

3 | METHODOLOGY

This section explains the combination of computer vision methods to provide a crack segmentation along with machine learning algorithms to perform the crack classification. In consequence, this section is divided into three main groups. The first group (subsection 3.1) details how to process the input images of road pavements to obtain the input features used by the machine learning algorithms and the proposed ensemble model. The second group (subsection 3.2) covers the learning algorithms used to classify each type of crack. The last group (subsection 3.3) details how to combine the previous machine learning algorithms into an ensemble model. Figure 1 illustrates the proposed methodology. Finally, the evaluation metrics to compare the different algorithms with the proposed model and other proposals are detailed in subsection 3.4.

3.1 | Feature extraction

The starting point of this approach is focused on data reduction in order to compute only useful information and to acquire the required inputs for the machine learning algorithms to classify different types of pavement deformations. The main steps of this block are detailed in Algorithm 1.

Algorithm 1 Feature extraction stage.

```
procedure FEATURE_EXTRACTION_METHOD(image)
    gray_img ← color_transformation (image)
    enhanced_img ← image_enhancement (gray_img)
    edge_img ← edge_detection (enhanced_img)
    modified_img←morphology_modifications (edge_img)
    New_features ← projections (modified_img)
    return New_features
end procedure
```

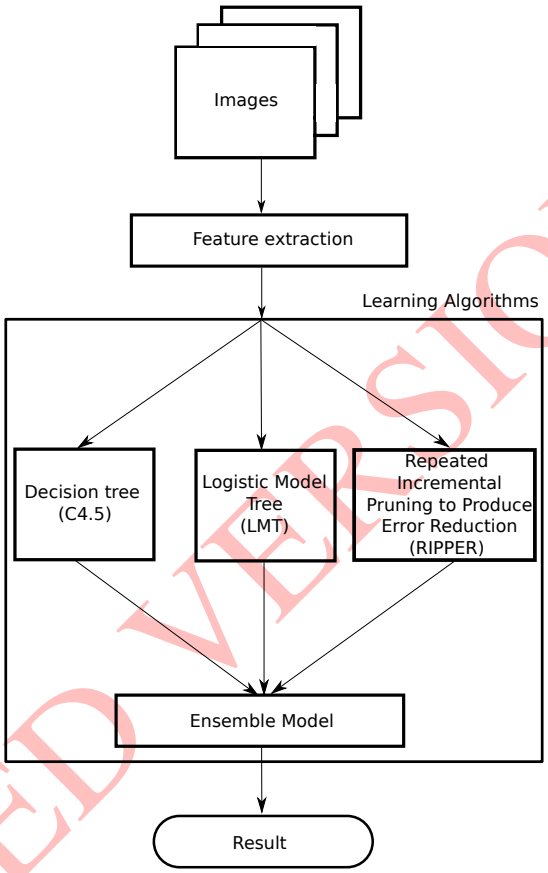

FIGURE 1 Proposed scheme.

The following subsections explain the purpose of each step of Algorithm 1.

3.1.1 | Color space transformation

The obtained images from a color camera are usually in RGB (Red–Green–Blue) space color with 8–bits per channel, as Figure 2 –a shows. However, there are three channels to represent a crack in the pavement and all colors are not needed. Hence, the RGB image is converted into a grayscale image (Kirk & Hwu, 2016).

This step also helps to mitigate the effect of any grass which may grow inside a crack (Figure 2 –a), and the effect of different illumination conditions (Figure 2 –b).

3.1.2 | Image enhancement

The next step is to enhance the grayscale image. This phase allows us to avoid false positives in future steps, applying contrast enhancement and a spatial filter.

The image enhancement is composed of three sequentially applied methods:

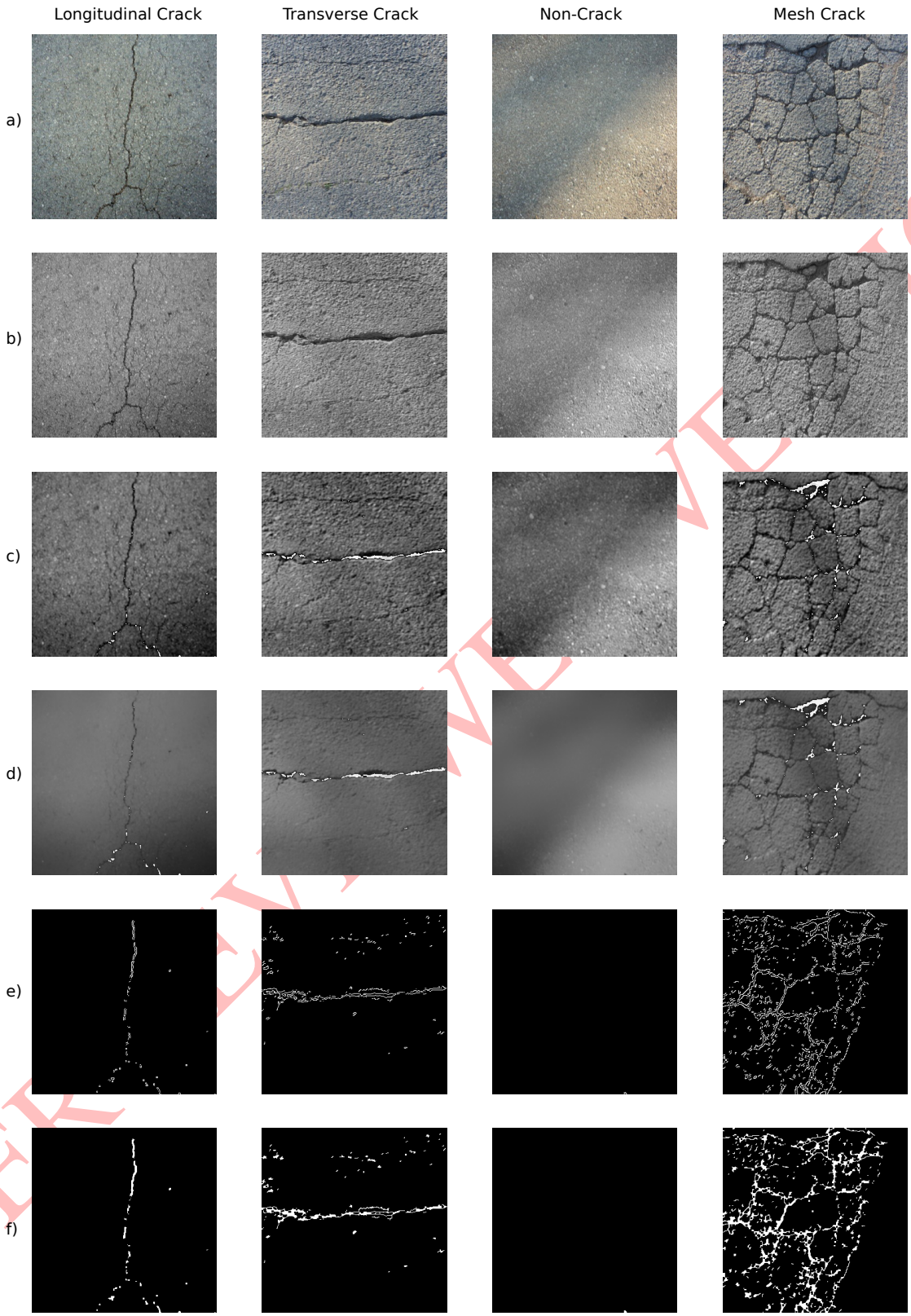

FIGURE 2 Results of image processing applied to Longitudinal crack, Transverse crack, Non–crack, and Alligator crack. a) Color images; b) Grayscale images; c) Logarithmic transformation; d) Bilateral filtering; e) Edge detection; f) Morphological closing.

- First, each pixel value in the grayscale image is replaced with its negative.
- The logarithmic transformation (Deng, Cahill, & Tobin, 1995) is applied to the negative image. This transformation achieves an improvement in the illumination of the image. It is possible to emphasize the cracks if they exist in the image and to attenuate the background. Two parameters (α and β) are required by the logarithmic transformation. Let β be the value to modify the sharpness of the image, and α the parameter used to change the dynamic range of the image. The values used in this research are shown in Section 4. The results of the negative and logarithmic transformation are shown in Figure 2 –c.
- Once the illumination has been improved, it is necessary to use a method for denoising the image keeping the main features of the crack, if this exists. Hence, it is not possible to use a linear filter to achieve this functionality. The Bilateral Filter (Tomasi & Manduchi, 1998) is applied because it is a non–linear filter which denoises the image without affecting the edges. Again, two parameters (σ_S and σ_r) are required by the bilateral filter. Let σ_S be the spatial parameter that controls the smoothing of the image and σ_r the range parameter which establishes the minimum amplitude of an edge. The values of these parameters are shown in Section 4. The result of the bilateral filter is shown in Figure 2 –d.

3.1.3 | Edge detection

Once the image has been enhanced, it is time to apply an edge detection algorithm to extract the main features of the cracks. The Canny method (Canny, 1987) is used to obtain the edges of the cracks. The Canny algorithm can be summarized in the following steps:

1. Gradient: calculating the derivatives in the x and y axes of the image using the Sobel filter (Sobel & Feldman, 1968).
2. Non–maximum suppression: the gradient produces edges that do not have maximum values and several false positive edges. These edges are suppressed.
3. Thresholding: in order to erase those elements that are not edges, a double threshold is used. A lower threshold th_1 is used to reject all pixels below this threshold. An upper threshold th_2 is used to accept all pixels with a value above this threshold.
4. Hysteresis: once the possible strong and weak edges have been detected, each edge is tracked. If an edge is not connected to a strong edge, the pixel is labeled as a false edge. The result of this algorithm is shown in Figure 2 –e.

These threshold values (th_1 and th_2) can be difficult to set manually. Hence, in order to automatically select the value of these parameters, the Otsu method (Yu, Dian-ren, Yang, & Lei, 2010) is applied.

The Otsu method splits the pixels of a grayscale image into two classes, and calculates the threshold value t which minimizes the weighted within–classes variance. This value is used to set the threshold values of the Canny method, where $th_2 = t$ and $th_1 = 0.5 \cdot t$.

3.1.4 | Morphological modifications

In the previous step, the majority of the edges in the image are detected. However, sometimes the edges are broken by several pixels, which were not detected by the Canny operator. To fix this, morphological operators (Davies, 2012) are used. These operators let us modify the shape of the binary edges of the previous step.

The closing operator, as Figure 2 –f shows, allows the connection of those pixels of a crack that are unconnected. This operator consists of two sequentially applied steps: a dilatation and an erosion.

The dilatation operator $\varphi(I)$ expands the edge content in the image according to Equation (1).

$$\varphi(I) = \{i + s | i \in I \wedge s \in S\} \tag{1}$$

Let i be a pixel of the image I and S a structure of a certain size (in this work a matrix of 3×3 elements centered in the middle, with all values set to 1). Assuming that a pixel to 1 represents an edge and a pixel to 0 does not represent an edge. Equation (1) expresses the following condition: if, at least, one pixel of S is equal to 1, pixel i is assigned a value of 1, otherwise, 0.

The opposite process of the dilatation $\varphi(I)$ is the erosion $\zeta(I)$. In this case, the edges of an image are contracted. Let us consider the same parameter set of the previous equation, the erosion operator is expressed by Equation (2).

$$\zeta(I) = \{i | \forall s \in S, i + s \in I\} \tag{2}$$

Where Equation (2) expresses the following condition: if all pixels of S are equal to 1, pixel i is assigned a value of 1, otherwise, 0.

3.1.5 | Projections

Once the cracks are represented by their edges, there are many 0 values that populate the image and which do not provide

useful information. Hence, in order to compress the data, projections are calculated. Vertical and horizontal projections allow us to represent the features of the crack with less information than the complete binary image. In fact, with only these two projections the cracks analyzed in this work (alligator, transverse and longitudinal cracks) can be completely described.

The vertical projection ($\Phi(I_j)$) of an image is calculated following Equation (3).

$$\Phi(I_j) = \sum_{i=1}^{C-1} I(i,j); \forall j = 0, ..., R-1 \quad (3)$$

Let $I(i,j)$ be a pixel of the image, i an iterator for the columns (represented by C), j the iterator for the rows of the image (represented by R).

Alternatively, let us consider the same parameter set of Equation (3), the horizontal projection ($\Theta(I_i)$) is calculated using Equation (4).

$$\Theta(I_i) = \sum_{j=1}^{R-1} I(i,j); \forall i = 0, ..., C-1 \quad (4)$$

At the end of this step, each image is represented as an instance $\{\Theta, \Phi\}$ which describes the main features of a crack, and it is used as the inputs for the different learning algorithms in the next section.

3.2 | Learning algorithms

This section covers the description of three machine learning algorithms based on decision trees and rule–based systems. These types of methods have been chosen based on the previous analyzed works in subsection 2.3. Hence, the next subsections explain the basic concepts of the decision trees using the C4.5 method (subsection 3.2.1), the Logistic Model Trees (LMT) method (subsection 3.2.2), and the rule induction method Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (subsection 3.2.3).

Notice that in order to feed the inputs of the learning algorithms every image is represented by $\{\Theta, \Phi, Class\}$. Let Θ and Φ be the output of the feature extraction step and $Class$ the label that corresponds to each type of crack in the range $[0; (numClass - 1)]$.

3.2.1 | Decision trees (C4.5)

Decision trees are machine learning algorithms which have been used in different scientific areas providing an analogy of human knowledge in the classification problem (Sharma & Kumar, 2016). One famous decision tree is The C4.5 algorithm (Quinlan, 1993)

The C4.5 is a method based on discrimination between classes. This algorithm manages two different type of nodes: leaves (which represent a class), and decision nodes (which are the elements that branch the tree). The decision tree is built using the depth-first strategy. The algorithm considers all possible "tests" that can divide the dataset and selects the test which results in the highest information gain (Quinlan, 1993). In this case, since the used projections to represent a crack are continuous values, a discretization of the attributes based on the minimum entropy (Quinlan, 1996) is applied to split the data to obtain discrete variables.

The main steps of the C4.5 algorithm are listed below:

1. If each instance belongs to the same class, the algorithm returns a leaf labeled by that class.
2. Compute for every attribute the gain information, and choose the best one based on the highest information gain value.
3. Create a decision node with the best–selected attribute in the previous step.
4. Splitting the instances based on the values of the selected attribute.
5. The above steps and the splitting process are repeated recursively until all the instances in a subset belong to the same class.

3.2.2 | Logistic model trees (LMT)

Logistic Model Trees (LMT) (Landwehr, Hall, & Frank, 2005) are a special type of tree that combines induction trees (Kotsiantis, 2013) with regression models. In fact, this algorithm has a standard decision tree structure with decision nodes (similar to the C4.5 algorithm) that split the data. The leaf nodes consist of logistic regression functions that allow the classification of each instance.

The logistic regression models are built using the LogitBoost algorithm (Landwehr et al., 2005). The number of iterations needed by the LogitBoost algorithm is calculated using a cross–validation approach (Stone, 1976). This number of iterations prevents training data from overfitting. LMT uses the CART cost–complexity metrics (Breiman, Friedman, Olshen, & Stone, 1984) to prune the tree impeding its growth.

The general steps of the LMT algorithm are the following:

- The root node (the first node of the tree) is divided into two groups. The division into two classes uses the LogitBoost algorithm.
- The child root nodes are then split using the LogitBoost algorithm. This step continues splitting all the child nodes until the C4.5 stopping criteria based on information gain is reached.

- Finally, the tree is generated using the pruning CART cost–complexity metric.

3.2.3 | Repeated Incremental Pruning to Produce Error Reduction (RIPPER)

Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (Cohen, 1995) is a rule induction algorithm (Langley & Simon, 1995) which uses a *divide–and–conquer* iterative method in order to build the rules. These rules are generated for each class from the least frequent class to the most frequent class. This feature is desirable when the training datasets are unbalanced.

The main steps of the RIPPER algorithm are explained below:

1. A loop iterates for each class from the minority class to the majority class.
2. Building stage: The training set is split into a growing set and a pruning set. This step has two tasks that are repeated until no more instances of a class are available in the growing set, or the error of the rules exceeds 50%.
   - New rules to classify a class are created by adding new conditions until a rule is 100% accurate. Every condition is selected using the gain information in the same way as in the C4.5 algorithm.
   - As a rule grows, the algorithm applies a prune, deleting those attributes that are unnecessary. The rule growth stops when it reaches 50% error. The error (Qabajeh, Thabtah, & Chiclana, 2015) is calculated as $2p/(p+n)$, where p are positive examples and n are negative examples.
3. Optimization stage: In this stage, two variants are generated for each rule. The first variant is generated from an empty rule and the second is generated by adding attributes to the original rule. The Decision Length metric (DL) (Cohen, 1995) is applied to calculate which rules are longer than others. The rule with the minimum DL value is selected. At the end of this stage, if there are any positive examples in the growing set, the Building stage is computed again.
4. Cleaning up stage: The DL is again calculated for the whole rule set, and all the rules that increase the DL are removed from the rule set.

3.3 | Ensemble models

Ensemble models are considered in the literature as committee–based learning classifiers as Zhou (2012) explains in his book.

Ensemble model mechanics can be seen as a moderator and the machine learning algorithms, as experts in a subject. A moderator asks the experts about a topic (classification into crack types). The final response is the result of the combination of each solution provided by the experts. The moderator may consider that the opinion of one expert is more relevant than the opinion of others. Therefore, the main advantage of ensemble models compared to only one learning algorithm is fault tolerance. For instance, if an expert makes an inaccurate prediction of a class, but the majority of the others make accurate predictions, then, the result of the ensemble model is an accurate prediction.

Hence, an ensemble is composed of a minimum of two models. These models can be different algorithms (heterogeneous ensemble) or the same algorithm with different models (homogeneous ensemble). In fact, there is no unique approach to build an ensemble model (Gomes, Barddal, Enembreck, & Bifet, 2017; Sagi & Rokach, 2018). Since there are multiple variants to build an ensemble model (Zhou, 2012), the majority voting approach is selected in this work. With majority voting it is possible to use existing trained models that provide an output without changing the ensemble algorithm. Theoretically, the probability of making the right decision in an ensemble model based on the majority voting is detailed in Equation (5) based on (Brown, Wyatt, Harris, & Yao, 2005).

$$ModelReliability = \sum_{x>\frac{m+1}{2}}^{m} \frac{m!}{(m-x)!\cdot x!}\cdot p^{x}\cdot(1-p)^{(m-x)} \quad (5)$$

Let p be the independent probability of making the right decision of a single component algorithm, m the number of learning algorithms that compose the ensemble to vote for a decision, and x the iterator of the summation (considering m is odd). Therefore, if Equation (5) is applied in the case of an ensemble of three models (voters), varying the probability of guessing from 0 to 1, it is possible to graphically represent the reliability of an ensemble as shown in Figure 3 . This figure shows in the x axis the probability of making the right decision (p) in intervals of 0.1, and the model reliability in the y axis with the same intervals. Also, this figure represents the reliability of a single model (dotted line) in comparison with the reliability of an ensemble model with three voters (continuous line). It can be seen that when the probability value of making the right decision is lower than 0.5 ($p < 0.5$), the ensemble model theoretically behaves worse than an individual model. However, when $p > 0.5$ the ensemble model improves the results compared to a simple model. This demonstrates theoretically that an ensemble model with three voters provides more accurate results than its components can provide individually.

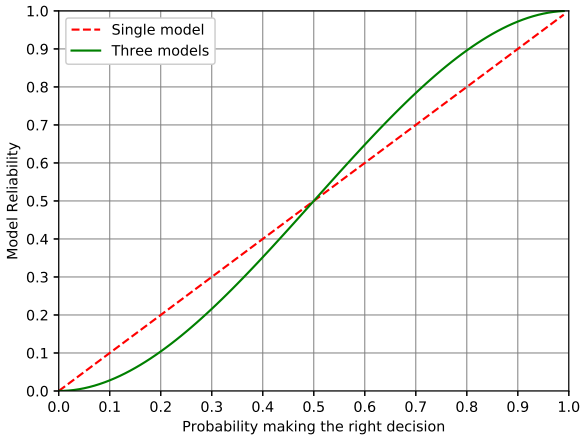

FIGURE 3 Reliability of a single model compared to an ensemble model with three components.

Algorithm 2 shows how to build the ensemble model with the outputs of the previous learning methods described in Section 3.2.

Algorithm 2 Ensemble model decision procedure.

```
1:  procedure MAJORITY_VOTING(Rla)
2:      R ← 0;                                  ▷ Class result.
3:      i ← 0;                                  ▷ Iterator.
4:      Class_number ← 4;                       ▷ Number of classes.
5:      CounterClass_i;        ▷ where 0 ≤ i < Class_number.
6:      while i < Class_number do
7:          for j=0: number of algorithms do
8:              if Rla_j equal to i then
9:                  CounterClass_i ← CounterClass_i + 1
10:             end if
11:         end for
12:     end while
13:     HighestVoted ← CounterClass_0;   ▷ Aux. variable.
14:     i ← 1
15:     while i < Class_number do
16:         if HighestVoted < CounterClass_i then
17:             R ← i
18:             HighestVoted ← CounterClass_i
19:         end if
20:     end while
21:     return R;
22: end procedure
```

The steps of Algorithm 2 are the following:

- Once each learning model has predicted its solution, it is stored in the variable called Rla.
- The classification labels are denoted by an integer from 0 to $Class\_number - 1$.
- For each class, a counter called $CounterClass$ is incremented if it is equal to the value of Rla for each algorithm.
- Once all the different results of each learning algorithm have been counted, a search to find the highest voted class is carried out.
- The class with highest score is assigned to the result (R) of the ensemble model. If all the voters have selected a different class for a pattern (no majority is achieved), the result of C4.5 is selected as the correct class, because it has been tested empirically that this algorithm obtains accurate results.

3.4 | Evaluation metrics

The computation of the performance of the learning algorithms is based on precision and recall metrics to fully evaluate the effectiveness of a model. These metrics are detailed in the following list:

- Precision (P): this measures the ratio between all predicted patterns of a class and the total number of patterns of that class. In other words, it is the relationship between True Positives (TP) and False Positives (FP), as Equation (6) shows:

$$\text{Precision (P)} = \frac{TP}{TP + FP} \tag{6}$$

- Weighted Average Precision (WAP): this represents precision (P) but regarding all the classes ($numClass$). This metric represents the precision of the algorithm. Equation (7) shows this formulation:

$$\text{WAP} = \sum_{i}^{numClass} \frac{n_i}{n_{total}} \cdot \frac{TP}{TP + FP} \tag{7}$$

Let n_i be the number of patterns in class i, and n_{total} the number of patterns in all the classes.

- Recall (R): this measures the instances of a class which are effectively classified in that class. It is the relationship between TP and True Negatives (TN), as Equation (8) shows:

$$\text{Recall (R)} = \frac{TP}{TP + TN} \tag{8}$$

- Weighted Average Recall (WAR): this represents the relationship between TP and TN considering all the classes. Equation (9) shows this formulation, where $numClass$, i, n_i, n_{total} are the same parameters as previously detailed:

$$\text{WAR} = \sum_{i}^{numClass} \frac{n_i}{n_{total}} \cdot \frac{TP}{TP + TN} \tag{9}$$

4 | EXPERIMENTS AND RESULTS

This section details the experiments carried out to analyze the performance of the proposed ensemble model detailed in Section 3.3. Hence, this section is divided into four subsections. Subsection 4.1 details the datasets used to carry out this work. Subsection 4.2 contrasts the results of the proposed ensemble model with its single components. Subsection 4.3 compares the proposed ensemble with the proposals of other authors. Finally, general considerations are stated in Subsection 4.4.

4.1 | Dataset composition and experiment settings

Three datasets were used in order to analyze the behavior with different patterns for the proposed ensemble of models and each of its components, as well as the proposals of other authors. The first dataset ('A') is composed of the images used in the work of Cubero-Fernandez et al. (2017). The second dataset ('B') is the combination of the previous dataset with 1856 new images collected from different road surfaces acquired for this work (each image was manually labeled as one of the analyzed types of cracks in this work). The last dataset ('C') is generated from dataset ('B'), performing data augmentation resulting in 19648 images. For all the labeled images, the number of patterns of each dataset is detailed in Table 1 . These images cover a wide number of cases and allow us to test the behavior of each algorithm with different data. Also, it can be observed for datasets 'B' and 'C' that the number of labeled images as non–cracks is higher. This is due to the fact that in real scenarios the number of cracks is lower than the non–crack patterns. This feature provides valuable information about how each algorithm is able to classify the minority classes. The dataset images are available on (The Advanced Informatics Research Group, 2019) for research purposes.

The parameters used by the computer vision algorithms (Section 3.1) have been selected empirically using 50 iterations in which the values of the parameters were changed, and validated the obtained results visually. The values for the logarithmic transformation are α: 0.9 and β: 1.1, with σ_S: 35 and σ_r: 16 for the bilateral filter.

TABLE 1 Dataset composition.

Data set	Alligator cracks	Longitudinal cracks	Transverse cracks	Non cracks
A	100	200	200	100
B	380	590	558	928
C	3040	4720	4464	7424

Regarding the design of the experiments a 10–fold cross–validation approach (Browne, 2000) was carried out for Table 2 , Table 3 and Table 4 . Also, the precision and recall metrics are computed by Equation (6) and Equation (8) respectively, to provide metrics about the performance in each class.

4.2 | Comparison with single models

This section contrasts the results of the different machine learning algorithms explained in Section 3.2 with the proposed ensemble model detailed in Section 3.3.

The results of the comparisons for each dataset are shown in Table 2 . This table shows that all the classification models are able to classify the different types of cracks with high precision values (No results below 80% are found). In the case of dataset 'A', LMT has the maximum precision for alligator cracks with a value of 94.9%. However, the reliability of a classification model cannot be based only on its precision. It is necessary to consider the recall metric to know how reliable the decisions of the generated models are. Therefore, although the proposed ensemble model provides a slightly lower precision result in the alligator crack type (94%), the recall has a value of 98%. This means that the ensemble model is slightly better in terms of recall and, thus, in terms of reliability than the LMT proposal. In the rest of classes for dataset 'A', the proposed ensemble method provides the highest scores.

For the case of dataset 'B', LMT provides slightly higher scores in alligator cracks in terms of recall than the proposed ensemble method (0.5% lower). However, in the rest of the classes (longitudinal, transverse, and non–cracks), the proposed model obtains the highest scores, as Table 2 shows. It can be observed that the class with the lowest values in general terms on precision and recall is the longitudinal cracks. This occurs because, unlike dataset A, these new data are more difficult to classify due to the presence of cracks with many arches in their path.

Finally, for dataset 'C' which corresponds to the augmented data, the proposed ensemble is less sensitive to the physical position of a crack in the image obtaining values up to 93%

in terms of recall and precision. Comparing the results for this dataset with the previous datasets, where the LMT algorithm produced the highest values of precision and recall, the proposed ensemble method improves the results in alligator cracks by 1.5%. In the case of longitudinal cracks this value increases in the ensemble model with a result 4.4% over LMT. Furthermore, the second most precise model is C4.5. For this reason, as was detailed in Section 3.3, it was taken as the most reliable result when there is no majority agreement between the voters.

Considering the last column of Table 2 , it can be observed that empirically, overall the proposed ensemble in terms of precision and recall provides more accurate results than the individual component algorithms for the three analyzed datasets, similar to the representation in Figure 3 .

4.3 | Comparison with other proposals

This section compares the performance of the proposed model to other proposals of Section 2, as Table 3 and Table 4 show. These were selected because they all use the same technology (2D images) and are focused on the classification of the same type of cracks analyzed in this work. Also, two of them use several classification models. In order to make a quantitative comparison, Equation (6) and Equation (8) are used to compute the precision and recall per class.

Table 3 contrasts our proposal with the works of Cubero-Fernandez et al. (2017) and Hoang & Nguyen (2018) for each dataset. As Table 3 shows, in the case of dataset 'A', our proposed ensemble model obtains higher scores in the classification of longitudinal cracks and non–cracks. The ensemble model increases the precision for longitudinal cracks by 31.5% and for transverse cracks by 25.6%, with respect to the results obtained by Cubero-Fernandez et al. (2017). However, for alligator cracks, the result is only 3% lower. These percentage increments are the effect of the combination of horizontal and vertical projection on the same instance, instead of separate projections to classify as in the proposal of Cubero-Fernandez et al. (2017). However, although the precision of the ensemble model for alligator cracks is slightly lower, the recall is 19.2% higher.

Analyzing the results of the proposal of Hoang & Nguyen (2018) for dataset 'A', it can be highlighted that compared to the use of a single learning model, the proposed ensemble model obtains improved results. In fact, our proposal shows (see Table 3) an increase of 10% in precision and 11% in recall when there are no cracks. For the rest of the types of cracks, the precision results of our proposal are higher than those of Hoang & Nguyen.

In the case of dataset 'B' our proposal provides higher classification results for transverse, longitudinal, and alligator crack types compared to the other authors. Only in the case of non–cracks is the proposed method lower in terms of precision compared to Hoang & Nguyen (2018) and in recall compared to Cubero-Fernandez et al. (2017). However, taking into consideration the remaining classes, it can be seen that the ensemble method is capable of maintaining results above 86.6%, which is not the case in the other proposals.

The proposals of Hoang & Nguyen (2018) and Cubero-Fernandez et al. (2017) show an increase of 1% and 2.5% respectively in terms of precision compared to our proposed method for dataset 'C'. However, as was detailed for the previous experiments the precision metric needs to be accompanied by the recall metric. In terms of this second metric, the proposed ensemble obtains the highest results. The only exception where the proposed method does not provide the highest results is in the classification of non–crack patterns where in terms of recall it is slightly lower (0.7%). In this case, although the recall is lower, the precision increases by up to 6.6% compared to Cubero-Fernandez et al. (2017). Hence, as Table 3 shows, the proposed ensemble (for the unbalanced datasets 'B' and 'C') is less sensitive to the majority and minority classes without penalizing the classification of the other classes.

The comparison with the proposal of L. Li et al. (2014) cannot be detailed in Table 3 because the number of types of cracks analyzed is not the same. Hence, Table 4 shows comparisons with their proposal. Since L. Li et al. (2014). do not classify the non–crack type, the patterns of this class were not used to make the comparison. For dataset 'A', in the classification of alligator cracks and linear cracks, both proposals work in the same way, achieving maximum precision and recall (100%). However, in the classification of the type of linear cracks into transverse cracks and longitudinal cracks, the ensemble model increases the precision by 16.3% and 10.9%, respectively. Also, the proposed model provides more accurate results for recall in the classification of transverse and longitudinal cracks, improving this metric by 17.5% and 10% respectively.

For datasets 'B' and 'C' where the classes are unbalanced the proposed ensemble model provides the highest precision and recall values in the classification of alligator cracks and linear cracks. The exception is in the case of the dataset 'B' where the proposed method obtains 0.5% less in the recall value. However, the precision in dataset 'B' is the highest. Analyzing the results with the patterns of longitudinal cracks and transverse cracks, the proposal of L. Li et al. (2014) obtains the highest values. This is probably because in dataset 'B' the features that represent the crack have connected components which are used as the inputs in the method used by L. Li et al. (2014). However, as observed in the case of dataset 'C', where cracks are spatially located in different places and components become disconnected, the ensemble provides better recall results for longitudinal cracks and better precision

TABLE 2 Results of the single models and the proposed ensemble model.

Set	Method	Alligator cracks		Longitudinal cracks		Transverse cracks		Non cracks		Weighted Average	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
A	C4.5	93.8%	92%	95.1%	94.5%	96.6%	96.5%	92.7%	94%	94.6%	94.2%
	LMT	**94.9%**	97%	93.6%	90%	92.1%	95%	99%	96%	94.9%	94.5%
	RIPPER	91.8%	91%	96%	94.5%	95.1%	96%	92.3%	92%	93.8%	93.4%
	Proposed model	94%	**98%**	**98.5%**	**95%**	**97.6%**	**99.5%**	**99.1%**	**97%**	**97.3%**	**97.4%**
B	C4.5	86.7%	86.8%	84.2%	82.7%	97.4%	98.5%	89.8%	86.8%	89.5%	89.1%
	LMT	89.1%	**91%**	83.6%	85.2%	97.4%	97.5%	91.1%	86.7%	90.3%	90.1%
	RIPPER	80.7%	89.5%	85.6%	82.4%	97.3%	98.4%	89.4%	89.1%	89.8%	89.8%
	Proposed model	**90.2%**	90.5%	**86.6%**	**87.4%**	**98.2%**	**98.5%**	**92.3%**	**89.4%**	**91.8%**	**91.5%**
C	C4.5	93.2%	92.3%	86.7%	83.6%	92.7%	92.2%	93.8%	96.6%	91.7%	91.8%
	LMT	92.9%	93%	84.5%	81.7%	91.7%	89.8%	93.5%	96.7%	90.8%	90.9%
	RIPPER	92.2%	92.3%	87.2%	81.6%	92.4%	92.8%	93.3%	96.8%	91.5%	91.5%
	Proposed model	**94.4%**	**94.6%**	**88.9%**	**85.9%**	**94.6%**	**94%**	**94.7%**	**97%**	**93.2%**	**93.3%**

TABLE 3 Comparisons with other authors, with complete datasets considering all types of cracks.

Set	Method	Alligator cracks		Longitudinal cracks		Transverse cracks		Non cracks	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
A	Cubero-Fernandez et al. (2017)	**97%**	78.8%	67%	81.7%	72%	81%	85%	80.1%
	Hoang & Nguyen (2018)	91.6%	98%	97.5%	**96%**	97%	96.5%	88.7%	86%
	Proposed model	94%	**98%**	**98.5%**	95%	**97.6%**	**99.5%**	**99.1%**	**97%**
B	Cubero-Fernandez et al. (2017)	61.28%	80.11%	80.89%	57.43%	91.77%	68.10%	81.47%	**98.20%**
	Hoang & Nguyen (2018)	90%	87.4%	76.3%	78.6%	89.7%	88.9%	**93.7%**	93.5%
	Proposed model	**90.2%**	**90.5%**	**86.6%**	**87.4%**	**98.2%**	**98.5%**	92.3%	89.4%
C	Cubero-Fernandez et al. (2017)	72.4%	84.3%	84.8%	70.6%	**97.1%**	85.5%	88.1%	**97.7%**
	Hoang & Nguyen (2018)	**95.4%**	92.8%	83%	79.3%	93%	91.9%	91.3%	95.6%
	Proposed model	94.4%	**94.6%**	**88.9%**	**85.9%**	94.6%	**94%**	**94.7%**	97%

TABLE 4 Comparisons with other authors, with reduced datasets without considering non–cracks.

Set	Method	Alligator cracks		Linear cracks		Longitudinal cracks		Transverse cracks	
		Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
A	L. Li et al. (2014)	100%	100%	100%	100%	86.6%	81%	82.2%	87.5%
	Proposed model	**100%**	**100%**	**100%**	**100%**	**97.5%**	**98.5%**	**98.5%**	**97.5%**
B	L. Li et al. (2014)	92%	69.5%	90.7%	**98%**	**98%**	**94.2%**	**94.8%**	**98%**
	Proposed model	**92.1%**	**88.9%**	**96.4%**	97.5%	91.9%	92.9%	92.4%	91.4%
C	L. Li et al. (2014)	91.1%	69.4%	90.6%	97.7%	**97.8%**	94.9%	94.8%	**97.7%**
	Proposed model	**95.9%**	**93.8%**	**97.9%**	**98.7%**	94.7%	**95.9%**	**95.6%**	94.3%

results for transverse cracks. The method of L. Li et al. (2014) provides more accurate values if only the two classes (longitudinal/transverse cracks) are analyzed separately. However, in the classification of non-crack patterns (generally the most common pattern) this method cannot classify them. Also, the recall for alligator cracks of their proposal obtains lower results.

4.4 | Research highlights

Once the behavior of the ensemble model is compared to the individual models (Subsection 4.2) and contrasted with other proposals (Subsection 4.3) for the different datasets, the following facts can be highlighted:

- The ensemble model provides better results on average in terms of accuracy and recall compared to the use of individual models.
- In the presence of datasets that include cases of majority and minority classes, the ensemble model is more sensitive, providing a classification for majority classes without penalizing minority classes as opposed to the other proposals.
- Combining models in an ensemble provides better results than multi–stage classification with the independent models proposed by other authors when all the classes are analyzed simultaneously.

5 | CONCLUSIONS

In this work, a method for the classification of non–cracks, longitudinal, transverse, and alligator cracks has been proposed. The method is composed of two stages: the first is based on computer vision algorithms to extract the fundamental characteristics of the cracks, and the second stage is the proposal of a new ensemble model based on decision trees and rule–based algorithms.

In terms of performance, the proposed ensemble model provides on average a crack classification higher than 94% with respect to the precision and recall metrics for the three analyzed datasets. Comparing the results to the individual components of our proposed ensemble (C4.5, LMT, and RIPPER), our approach provides in general, the highest values of precision and recall for the analyzed classes in each dataset. Also, the ensemble provides the highest weighted average precision and recall values.

In comparison with other authors, our model obtains in general, the highest values in precision and recall when all the classes are analyzed simultaneously. In those cases where one of these two metrics is slightly lower, the other metric is the highest. In the case of binary classification (alligator/linear crack classification) for the largest analyzed dataset, the improvement in results of our proposal is up to 24.4% and 4.8% better in terms of recall and precision respectively than the proposals of other authors.

Finally, our proposal has demonstrated that the use of ensemble models, taking as individual experts decision trees and rule–based algorithms, can work effectively in the problem of crack classification producing promising results. Also, our proposal mitigates the effect of miss–classifications when the datasets are unbalanced. Hence, it opens a new world of possibilities for the combination of different classification models.

6 | FUTURE RESEARCH

This work has focused on the classification of the most common types of cracks in pavement road surfaces. To achieve this, computer vision for road crack segmentation and an ensemble model was proposed. However, there are some issues that could be addressed in future research. Firstly, the analyzed cracks in this work could be expanded with new types of defects, such as viscoplastic deformations (for instance, potholes) or surface defects (such as skidding effect or aggregate pollution (Ragnoli et al., 2018)). Secondly, it would be interesting to compare the road crack segmentation with proposals based on deep learning. Also, the incorporation of new learning algorithms to exploit the advantage of the ensembles in terms of scalability and adaptability (Sun et al., 2013) could be addressed.

Finally, this work is exclusively focused on road pavements. However, it would be interesting to explore the research presented in this paper for use in different civil engineering scenarios in which cracks may appear, such as tunnels or concrete structures.

ACKNOWLEDGMENTS

This work has been supported by the Advanced Informatics Research Group – GIIA (TIC-252) Universidad de Córdoba (Spain).

Author contributions

The contributions of each author are the following: Francisco J. Rodriguez-Lozano is the main author who conceived and designed the experiments and wrote the paper. Fernando León-García and Juan C. Gámez-Granados helped to analyze the data. Finally, Jose M. Palomares and J. Olivares are the chief researchers who supervised all the work, reviewed the article, and helped throughout the process. All authors have read and approved the final manuscript.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

References

Ai, D., Jiang, G., Kei, L. S., & Li, C. (2018). Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods. *IEEE Access*, *6*, 24452–24463.

Arage, S. S., & Dharwadkar, N. V. (2017). Cost estimation of civil construction projects using machine learning paradigm. In *2017 international conference on i-smac (iot in social, mobile, analytics and cloud) (i-smac)* (p. 594-599).

Babashamsi, P., Yusoff, N. I. M., Ceylan, H., Nor, N. G. M., & Jenatabadi, H. S. (2016). Evaluation of pavement life cycle cost analysis: Review and analysis. *International Journal of Pavement Research and Technology*, *9*(4), 241–254.

Bang, S., Park, S., Kim, H., san Yoon, Y., & Kim, H. (2018). A deep residual network with transfer learning for pixel-level road crack detection. In *Proceedings of the 35th international symposium on automation and robotics in construction (ISARC)*. International Association for Automation and Robotics in Construction (IAARC).

Bella, F., Calvi, A., & D'Amico, F. (2012). Impact of pavement defects on motorcycles' road safety. *Procedia - Social and Behavioral Sciences*, *53*, 942–951.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. belmont, ca: Wadsworth. *International Group*, 432.

Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, *6*(1), 5 - 20. Diversity in Multiple Classifier Systems.

Browne, M. W. (2000). Cross-validation methods. *Journal of Mathematical Psychology*, *44*(1), 108 - 132.

Canny, J. (1987). A computational approach to edge detection. *Readings in Computer Vision*, 184–203.

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Chai, G., van Staden, R., Guan, H., Kelly, G., & Chowdhury, S. (2016). The impacts of climate change on pavement maintenance in queensland, australia. In *Materials and infrastructures 2* (pp. 207–221). John Wiley & Sons, Inc.

Cheng, J., Xiong, W., Chen, W., Gu, Y., & Li, Y. (2018). Pixel-level crack detection using u-net. In *TENCON 2018 - 2018 IEEE region 10 conference*. IEEE.

Cohen, W. W. (1995). Fast effective rule induction. In A. Prieditis & S. Russell (Eds.), *Machine learning proceedings 1995* (p. 115 - 123). San Francisco (CA): Morgan Kaufmann.

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.

Davies, E. (2012). Chapter 7 - mathematical morphology. In E. Davies (Ed.), *Computer and machine vision (fourth edition)* (Fourth Edition ed., p. 185 - 208). Boston: Academic Press.

Deng, G., Cahill, L., & Tobin, G. (1995). The study of logarithmic image processing model and its application to image enhancement. *IEEE Transactions on Image Processing*, *4*(4), 506–512.

Foss, K. (2017). Welcome to your autonomous life: Self driving cars are a new reality. *Science Trends*.

Galbraith RM, P. D. (2005). *Scottish road network climate change study*. The Scottish Government, Glasgow.

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

Gomes, H. M., Barddal, J. P., Enembreck, F., & Bifet, A. (2017). A survey on ensemble learning for data stream classification. *ACM Comput. Surv.*, *50*(2), 23:1–23:36.

Gutierrez Soto, M., & Adeli, H. (2017). Recent advances in control algorithms for smart structures and machines. *Expert Systems*, *34*(2), e12205.

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*.

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.

James, A. P. (2016). Edge detection for pattern recognition: a survey. *International Journal of Applied Pattern Recognition*, *3*(1), 1.

Kim, H., Ahn, E., Shin, M., & Sim, S.-H. (2018). Crack and noncrack classification from concrete surface images using machine learning. *Structural Health Monitoring*, *18*(3), 725–738.

Kirk, D. B., & Hwu, W.-m. W. (2016). *Programming massively parallel processors, third edition: A hands-on approach* (3rd ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*(4), 261–283.

Landwehr, N., Hall, M., & Frank, E. (2005). Logistic model trees. *Machine Learning*, *59*(1-2), 161–205.

Langley, P., & Simon, H. A. (1995). Applications of machine learning and rule induction. *Commun. ACM*, *38*(11), 54–64.

Lee, J., Nam, B., & Abdel-Aty, M. (2015). Effects of pavement surface conditions on traffic crash severity. *Journal of Transportation Engineering*, *141*(10), 04015020.

Li, B., Wang, K. C. P., Zhang, A., Yang, E., & Wang, G. (2018). Automatic classification of pavement crack using deep convolutional neural network. *International Journal of Pavement Engineering*, 1–7.

Li, L., Sun, L., Ning, G., & Tan, S. (2014). Automatic pavement crack recognition based on bp neural network. *Promet Traffic&Transportation*, *26*(1), 11-22.

Li, Z., Cheng, C., Kwan, M.-P., Tong, X., & Tian, S. (2019). Identifying asphalt pavement distress using UAV LiDAR point cloud data and random forest classification. *ISPRS International Journal of Geo-Information*, *8*(1), 39.

Long, Y. (2008). Adaptive image enhancement algorithm based on bandelet transform. *Journal of Computer Applications*, *28*(5), 1221–1224.

Luo, W., Wang, K. C. P., Li, L., Li, Q. J., & Moravec, M. (2014). Surface drainage evaluation for rigid pavements using an inertial measurement unit and 1-mm three-dimensional texture data. *Transportation Research Record: Journal of the Transportation Research Board*, *2457*(1), 121–128.

Mills, B. N., Tighe, S. L., Andrey, J., Smith, J. T., & Huen, K. (2009). Climate change implications for flexible pavement design and performance in southern canada. *Journal of Transportation Engineering*, *135*(10), 773–782.

Nayyeri, F., Hou, L., Zhou, J., & Guan, H. (2018). Foreground–background separation technique for crack detection. *Computer-Aided Civil and Infrastructure Engineering*, *34*(6), 457–470.

Prayogo, D., Cheng, M.-Y., Wu, Y.-W., & Tran, D.-H. (2019). Combining machine learning models via adaptive ensemble weighting for prediction of shear capacity of reinforced-concrete deep beams. *Engineering with Computers*.

Qabajeh, I., Thabtah, F., & Chiclana, F. (2015). A dynamic rule-induction method for classification in data mining. *Journal of Management Analytics*, *2*(3), 233–253.

Qiao, Y., Chen, S., Alinizzi, M., & Labi, S. (2018). Modeling the relationships between pavement distress and performance. In *Advances in materials and pavement performance prediction* (pp. 11–13). CRC Press.

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

Quinlan, J. R. (1996). Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, *4*, 77–90.

Radopoulou, S. C., & Brilakis, I. (2017). Automated detection of multiple pavement defects. *Journal of Computing in Civil Engineering*, *31*(2), 04016057.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400.

Rafiei, M. H., & Adeli, H. (2018a). Novel machine-learning model for estimating construction costs considering economic variables and indexes. *Journal of Construction Engineering and Management*, *144*(12), 04018106.

Rafiei, M. H., & Adeli, H. (2018b). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2).

Ragnoli, A., Blasiis, M. D., & Benedetto, A. D. (2018). Pavement distress detection methods: A review. *Infrastructures*, *3*(4), 58.

Ronneberger, O., P.Fischer, & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention (miccai)* (Vol. 9351, pp. 234–241). Springer.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1249.

Schweikert, A., Chinowsky, P., Espinet, X., & Tarbert, M. (2014). Climate change and infrastructure impacts: Comparing the impact on roads in ten countries through 2100. *Procedia Engineering*, *78*, 306–316.

Shang, W., Huang, H., Zhu, H., Lin, Y., Wang, Z., & Qu, Y. (2005). An improved kNN algorithm – fuzzy kNN. In *Computational intelligence and security* (pp. 741–746). Springer Berlin Heidelberg.

Sharma, H., & Kumar, S. (2016). A survey on decision tree algorithms of classification in data mining. *International Journal of Science and Research (IJSR)*, *5*(4), 2094–2097.

Sobel, I., & Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, 271–272.

Song, Q., & Cosman, P. C. (2018). Luminance enhancement and detail preservation of images and videos adapted to ambient illumination. *IEEE Transactions on Image Processing*, *27*(10), 4901–4915.

Soto-Hidalgo, J. M., Alonso, J. M., Acampora, G., & Alcala-Fdez, J. (2018). JFML: A java library to design fuzzy logic systems according to the ieee std 1855-2016. *IEEE Access*, *6*, 54952-54964.

Souza, V. M. (2018). Asphalt pavement classification using smartphone accelerometer and complexity invariant distance. *Engineering Applications of Artificial Intelligence*, *74*, 198 - 211.

Spencer, B. F., Hoskere, V., & Narazaki, Y. (2019). Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering*, *5*(2), 199–222.

Stone, M. (1976). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society: Series B (Methodological)*, *38*(1), 102–102.

Sun, J., Li, H., & Adeli, H. (2013). Concept drift-oriented adaptive and dynamic support vector machine ensemble with time window in corporate financial risk prediction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *43*(4), 801-813.

The Advanced Informatics Research Group. (2019). *Road crack dataset.* Retrieved 2019-12-15, from `https://www.uco.es/giia/road-crack-dataset/`

The World Health Organization. (2018). *Global status report on road safety 2018.* Retrieved 2019-06-09, from `https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/`

Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (ieee cat. no.98ch36271)* (p. 839-846).

Wang, B., Li, Y., Zhao, W., Zhang, Z., Zhang, Y., & Wang, Z. (2019). Effective crack damage detection using multilayer sparse feature representation and incremental extreme learning machine. *Applied Sciences*, *9*(3), 614.

Wang, T., Gopalakrishnan, K., Smadi, O., & Somani, A. K. (2018). Automated shape-based pavement crack detection approach. *Transport*, *33*(3), 598–608.

Xu, G., Chen, F., Wu, G., & Li, X. (2018). Active solution of homography for pavement crack recovery with four laser lines. *Scientific Reports*, *8*(1).

Yu, C., Dian-ren, C., Yang, L., & Lei, C. (2010). Otsu's thresholding method based on gray level-gradient two-dimensional histogram. In *Proceedings of the 2nd international asia conference on informatics in control, automation and robotics - volume 3* (pp. 282–285). Piscataway, NJ, USA: IEEE Press.

Zhou, Z.-H. (2012). *Ensemble methods: Foundations and algorithms* (1st ed.). Chapman & Hall/CRC.

How to cite this article: Francisco J. Rodriguez-Lozano, Fernando León-García, Juan C. Gámez-Granados, Jose M. Palomares, J. Olivares, Benefits of ensemble models in road pavement cracking classification. *Comput Aided Civ Inf*, .