10

Supervised Crack Detection and Classification in Images of Road Pavement Flexible Surfaces

Henrique Oliveira[1,2] and Paulo Lobato Correia[1]
[1]*Instituto de Telecomunicações – Instituto Superior Técnico, Univ. Técnica de Lisboa*
[2]*Escola Superior de Tecnologia e Gestão – Instituto Politécnico de Beja*
Portugal

1. Introduction

Roads are important man-made infrastructures that exhibit distresses due to their constant usage, its maintenance being an essential task to ensure a correct pavement performance. To this end, good road maintenance policies are required, relying on adequate rehabilitation management procedures. Periodic road pavement surveys provide the necessary collection of data about pavement surface condition, an essential tool to decide on the appropriate maintenance techniques to be applied in pavement rehabilitation (restoration and repairs).

During periodic road pavement surveys, human visual inspection aims to provide a good structural and functional quality evaluation. This type of inspection is traditionally done by skilled technicians (inspectors) that travel along the surveyed road, acquiring images (the most important source of information for quantitative and qualitative distresses evaluation) and annotating them about pavement surface distresses types (cracks and other degradations) and their location.

The analysis of distresses relies on the inspector's experience on pavement surface observation. Some major drawbacks of this procedure are (Meignen et al., 1997; Chen & Miyojim, 1998): it is labor-intensive, since usually less than 10km per day can be surveyed and a considerable effort is required from those skilled technicians to manually analyze the full set of acquired images; it is prone to subjectivity, since two inspectors can produce different analysis results over similar distress situations. An automatic crack detection system based on the analysis of road pavement surface images, as proposed here, can significantly speed up the process and reduce results' subjectivity.

The methodologies reported so far, mainly deal with automatic crack detection and classification using techniques such as neural networks, including fuzzy sets and the computation of moment invariants, Markov random fields and edge detectors, among others, as discussed in Section 2. This text explores a novel approach for automatic crack distresses detection and classification in images taken over road pavement flexible surfaces, relying on image processing and pattern recognition techniques, envisaging a simple framework based on local statistics computed over non-overlapping image regions.

For the automatic detection of crack distresses (the first task of the proposed system), this chapter describes supervised classification strategies, composed of two main steps: training

and testing. To supply adequate input data to these steps, the entire image database is split into two subsets, resorting to an automatic selection of training images. The training image subset is used for classifier training, in which a human expert manually selects image regions containing crack pixels. The test image subset is composed by images to be automatically classified, here used also for crack detection evaluation.

All the images are pre-processed using normalization techniques to reduce the effects of non-uniform illumination and to mitigate the influence of noise and some irrelevant properties of pavement upper layer aggregates. In training and test steps, six supervised classification strategies, three parametric and three non-parametric, are confronted to infer about their crack detection suitability (Oliveira & Correia, 2008). These classification strategies explore a 2D feature space, relying on the mean value of pixel intensities in non-overlapping image regions, together with the corresponding standard deviations. The 2D feature space is subsequently normalized to reduce feature value scattering among database images, leading to a better classifier performance.

The results obtained in the crack detection task are subsequently used as input for a crack type classification task, where images are labeled as containing longitudinal, transversal and miscellaneous or no cracks, a subset of the crack distress types identified in the Portuguese Distress Catalogue (JAE, 1997). This second task explores another 2D feature space, computed for this purpose.

The proposed automatic system is evaluated over an image database composed by real flexible pavement surface images, acquired during a survey over a Portuguese road (following the visual inspection method), using a set of well-know metrics and exploiting the availability of ground truth data manually provided (human labeling) for the entire image database. Promising results are obtained in both crack detection and classification tasks.

This chapter is structured as follows. Section 2 reviews the most relevant literature addressing automatic crack detection and classification systems. Section 3 discusses image acquisition, the automatic selection of training images, image normalization and saturation, followed by feature extraction and normalization. Section 4 describes the classification strategies used for crack detection and the associated parameters while Section 5 presents the approach for crack distress type classification. Section 6 discusses the experimental results. Conclusions and some hints for future work are addressed in Section 7.

2. Literature Review

In the scientific literature, not so many papers dealing with the automatic detection of pavement surface distresses are available. A good starting point can be found in (Chen & Miyojim, 1998), which reviews the techniques applied for the development of automatic pavement distress detection and classification systems. They also propose a novel approach according to the following major steps: region based image enhancement, to correct nonuniform background illumination and a skeleton analysis algorithm to classify pavement surface distress types. Experimental results for forty two crack samples are analyzed, corresponding to six different distress types, notably transversal, diagonal, longitudinal, block, combination and alligator pattern cracks. They claim 100% accuracy in the distress type classification results, but no evaluation of the crack detection step is included.

(Chou et al., 1994) proposed the use of moment invariant features and trained neural networks, again focusing on crack type classification. The same crack distress types of (Chen & Miyojim, 1998) are considered plus right diagonal, left diagonal non-distress types. The authors claim 100% accuracy for crack distress type classification, but crack detection results are not reported.

A multi-scale approach using Markov Random Fields for crack detection is presented in (Chambon et al., 2009). Cracks are enhanced using a Gaussian function and then processed by a 2D matched filter to detect cracks. Another approach, based on a nonsubsampled contourlet transform for pavement distress crack detection, is proposed in (Ma et al., 2008) but very few experimental results are provided.

An artificial living system is proposed in (Zhang & Wang, 2004). This work includes a pre-processing step to exclude bright points in images and enhance the contrast between crack and non-crack pixels (based on top-hat and bottom-hat procedures). No ground truth information is provided here and experimental results are mainly qualitative.

The use of edge detectors for pavement distress evaluation is proposed in (Li et al., 1991), while a pavement image segmentation technique is presented in (Qingquan & Xianglong, 2008) to identify image cracks. A novel thresholding technique is used, the authors claiming to achieve better experimental results than with well-known thresholding methods. Another thresholding-based technique is presented in (Liu et al., 2008), where the resulting binary images are processed by a connected domain algorithm to find cracks. No information about the use of ground truth information is provided.

Also the problem of pavement image acquisition and the related hardware is discussed in some papers. (Wang, 2000) describes new hardware acquisition systems as well as their applicability to the automation of pavement distress evaluation. Also (Huang & Xu, 2006) describes a system where images are acquired using a time delay integration line-scan camera.

3. Crack Detection Supervised Strategies

The proposed system architecture for automatic crack detection and classification is shown in Fig. **1**. The first step aims to split the image database into two subsets: the Training Image Set (TIS), used to train classifiers with manually labeled samples (image regions) containing crack pixels; the Testing Image Set (TTIS), the remaining images to be automatically processed by the system for crack detection and crack types classification. A set of procedures (image normalization and saturation and feature normalization) are implemented to ensure the computation of adequate decision boundaries, aiming better crack detection performance.

Two main tasks are identified: detection, where image regions are labeled as containing crack pixels or not; and crack type classification, where 'longitudinal', 'transversal', 'miscellaneous' or 'no cracks' labels are assigned to each detected crack.

Detailed discussion on training images selection, feature extraction, normalization (for both image and feature space data) and saturation procedures is given in the following sub-sections.

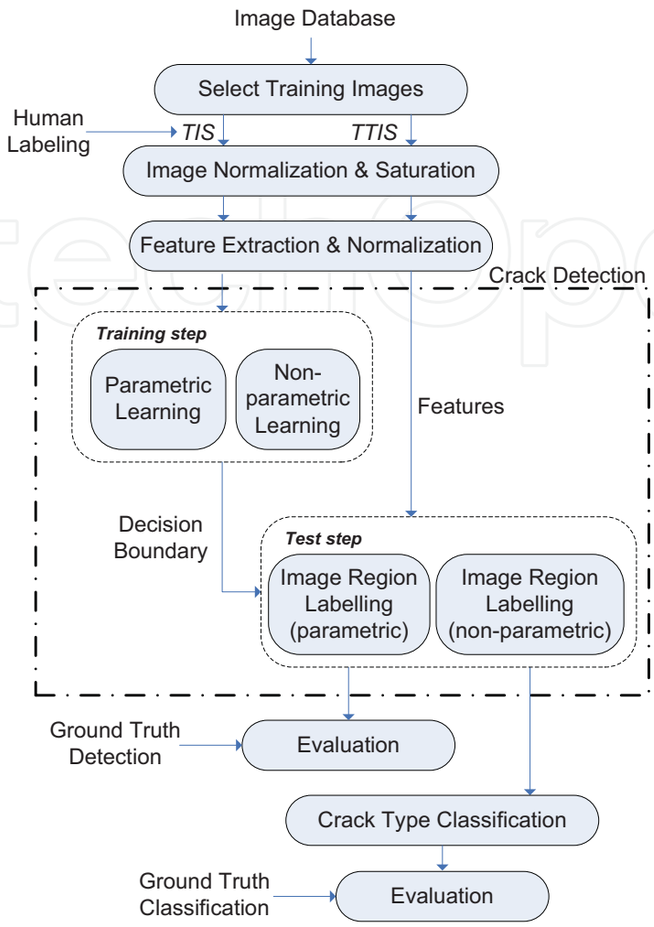


Fig. 1. System architecture.

3.1 Image Acquisition

The image database considered in this research work is composed by grayscale images, acquired during a pavement surface visual survey over a Portuguese road. A digital camera was manually positioned by the inspector with its optical axis perpendicular to the road surface, at a distance of approximately 1.2 m. Images with different sizes are obtained (2048×1536 pixels and 1858×1384 pixels), according to different camera setup procedures. The digital camera is oriented in such a way that the images only contain areas belonging to the road pavement surface. Moreover, the database includes images with several types of cracks (longitudinal, transversal and miscellaneous), as well as images without any cracks.

Instead of processing the images at a pixel level in all the steps of the proposed system, each image is divided into a set of non-overlapping regions of size 75×75 pixels. These dimensions were empirically chosen, leading to a faster processing time and lower memory storage requirements, while providing a good compromise between complexity and accuracy. Database images can then be represented by smaller matrices, where each of their values corresponds to the computation of region local statistics, as described next.

3.2 Selection of Training Images

Dealing with supervised classification strategies, training data (images for the envisaged application) is necessary for classifiers learning. This section describes a technique for the automatic selection of images, to be included in TIS, from the entire image database acquired during the visual road pavement survey.

To allow a correct learning stage, training images should contain road pavement cracks. Therefore, in a preliminary classification phase, all images are pre-processed in order to detect the regions with most evident crack pixels, by exploiting the knowledge that regions with crack pixels are supposed to have lower average intensities, when compared to regions without crack pixels. The images are then sorted, starting from those where the longest cracks were detected, the TIS being chosen from the top of this sorted list. The number of images to be included in TIS is an option controlled by the system operator. Moreover, the operator can edit the TIS, i.e., he can manually reject images automatically labeled by the system as 'training image' or add additional ones. Images definitely labeled as 'training images' are finally presented to the system operator, for manual identification of regions containing crack pixels.

In this preliminary classification phase, image regions revealing evident crack pixels are automatically labeled '1', or '0' otherwise. The result is a binary matrix (M_{bm}) with dimensions nl_{bm} and nc_{bm}, given by:

$$nl_{bm} = fix\left(\frac{nl_{img}}{nl_r}\right) \text{ and } nc_{bm} = fix\left(\frac{nc_{img}}{nc_r}\right) \quad (1)$$

where nl_{img} and nc_{img} stand for the number of lines and columns of an image, respectively; nl_r and nc_r are the number of lines and columns of regions (here square regions of 75x75 are used, as referred in Section 3.1), and *fix* is an operator which rounds a number towards zero.

Automatic image region labeling, in the preliminary classification phase, starts with the computation of a regions' mean values matrix - M_{rm}, with dimensions $nl_{bm} \times nc_{bm}$, each of its elements representing the region's pixel intensities average. This matrix is vertically and horizontally scanned to find regions with evident crack pixels, by analyzing the variation of the average region values when compared to those of the nearest neighbors, also taking into account all the values along the line or column under analysis.

Starting with the vertical scanning of M_{rm}, a region is considered a candidate of containing cracks when the following logical decision, $ld^{(V)}$, holds true:

$$ld^{(V)} = \left[\text{std}(\text{Av}^{(\text{i,j})}) > k_1 \times \text{std}(\text{Bv}^{\text{j}}) + k_2 \times \text{mean}(\text{Bv}^{\text{j}})\right] \wedge \left[\left(\text{Av}^{(\text{i,j})}[1] - \text{Av}^{(\text{i,j})}[2]\right) > 0\right] \quad (2)$$

with

$$\text{Av}^{(\text{i,j})} = \begin{bmatrix} \dfrac{rm_{(\text{i}-1,j)} + rm_{(\text{i}+1,\text{j})}}{2} \\ rm_{(\text{i,j})} \end{bmatrix}, \ \text{Bv}^{\text{j}} = \begin{bmatrix} 0 \\ \text{std}\left(\text{Av}^{(2,\text{j})}\right) \\ \vdots \\ \text{std}\left(\text{Av}^{(nl_{bm}-1,\text{j})}\right) \\ 0 \end{bmatrix}, \quad (3)$$

where $rm_{(\text{i,j})}$ corresponds to the average pixel intensity of a region at position (i,j), k_1 and k_2 are parameters controlled by the system operator (set by default to an empirically chosen value) and $\text{Av}^{(\text{i,j})}$ and Bv^{j} are column vectors with dimensions 2×1 and nlbm×1, respectively. Elements of Bv^{j} represent the standard deviation between region average intensities along row *i* and column *j* (i.e. rm(i,j)) and the corresponding values of its nearest vertical

neighboring regions ($[rm_{(i-1,j)} + rm_{(i+1,j)}]/2$). Bv^j is used to gather some knowledge about the expected variations along the columns of M_{rm}, highlighting the presence of relevant dark pixels in regions, to be accounted for in equation (2). Regions with relevant crack pixels have higher std(Bv^j) values, due to higher $Av^{(i,j)}$ values when compared to regions without crack pixels. Additionally, the values of $Av^{(1,j)}$ and $Av^{(nl_{bm},j)}$, i.e. the extreme regions of each column (top and bottom edges), take value zero. After the vertical scanning of M_{rm}, a binary matrix, $M_{bm}^{(V)}$, is build with the computed $ld^{(V)}$ values; it has the same dimensions of M_{rm}.
Fig. **2** is used to illustrate the behavior of std(Bv^j) in the presence of cracks. It shows a sample column of Mrm matrix (12th column) in two road pavement surface images. The std(Bv^j) value computed for the regions of the left image is lower (0.5696) than the corresponding value for the right image (1.1895), due to the existence of an higher std($Av^{(11,12)}$) value when compared to std($Av^{(i,12)}$) for the remaining regions. The same tendency is observed for mean(Bv^j), presenting a lower value for the left image (0.9405) than for the right image (1.3788).

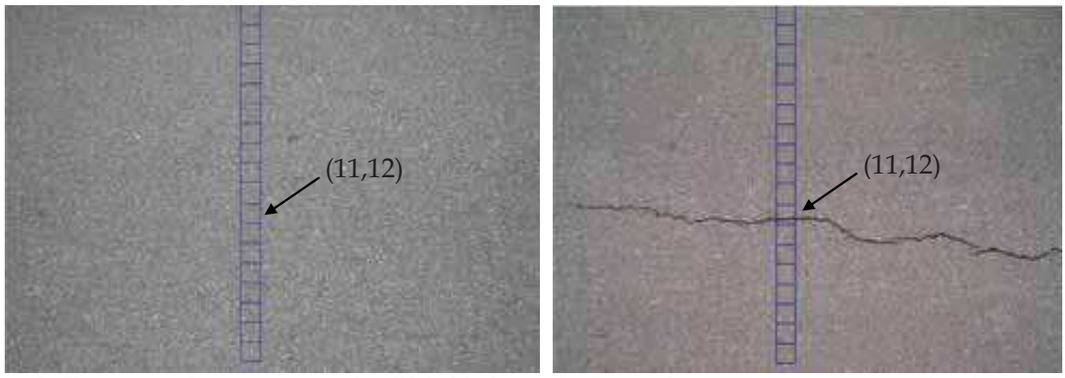


Fig. 2. Two sample images, with 1536x2048 pixels, from the pavement survey database. The left image shows a pavement surface without cracks, while the right image includes a transversal crack. Processed 75x75 pixel regions are marked with squares.

After the vertical scan, a horizontal scan proceeds in a similar way, acquainting for longitudinal cracks, which would be difficult to detect in a vertical scan. Expressions (4) and (5), for the horizontal scan, are similar to (2) and (3), with Av and Bv being replaced by Ah and Bh, respectively:

$$ld^{(H)} = \left[\text{std}(\text{Ah}^{(i,j)}) > k_1 \times \text{std}(\text{Bh}^i) + k_2 \times \text{mean}(\text{Bh}^i)\right] \wedge \left[\left(\text{Ah}^{(i,j)}[1] - \text{Ah}^{(i,j)}[2]\right) > 0\right] \quad (4)$$

$$\text{Ah}^{(i,j)} = \left[\frac{rm_{(i,j-1)} + rm_{(i,j+1)}}{2};\quad rm_{(i,j)}\right],\ \text{Bh}^i = \left[0;\quad \text{std}\left(\text{Ah}^{(i,2)}\right);\quad \ldots\quad \text{std}\left(\text{Ah}^{(i,nc_{bm}-1)}\right);\quad 0\right], \quad (5)$$

with $Ah^{(i,j)}$ and Bh^i being vectors with dimensions 2×1 and ncbm×1, respectively, and the values for $Ah^{(i,1)}$ and $Ah^{(i,nc_{bm})}$, i.e. the extreme regions of each row (left and right edges), taking value zero. After the horizontal scanning of M_{rm}, a new binary matrix with the computed $ld^{(H)}$ values is build, $M_{bm}^{(H)}$ (with the same dimensions of M_{rm}).

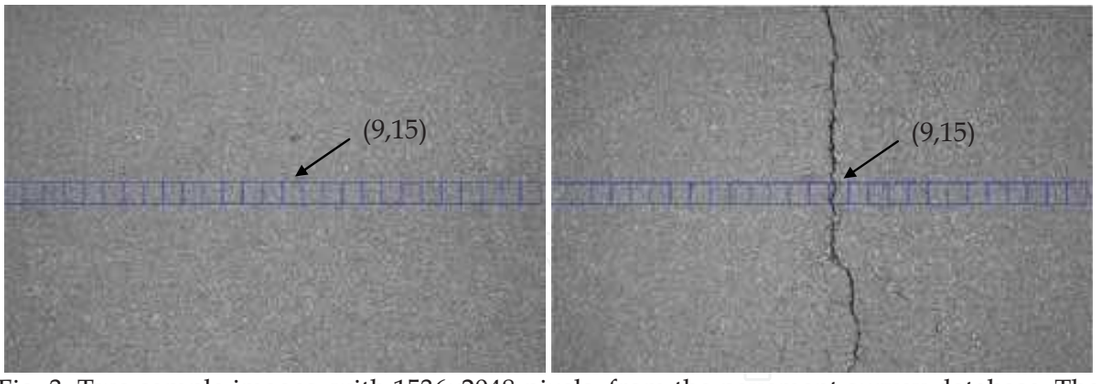


Fig. 3. Two sample images, with 1536x2048 pixels, from the pavement survey database. The left image shows a pavement surface without cracks, while the right image includes a longitudinal crack. Processed 75x75 pixel regions are marked with squares.

As an example, a horizontal scanning for the Mrm matrix 9th row of the images in Fig. **3** is considered. Lower values for std(Bhi) and mean(Bhi) are obtain for the left image (0.6002 and 1.0681, respectively) than for the right image (0.9298 and 1.2171, respectively), due to the existence of an higher std($Ah^{(9,15)}$) value when compared to std($Ah^{(9,j)}$) of the remaining regions.

The next step of the preliminary detection of regions containing cracks is to merge the two binary matrices $M_{bm}^{(V)}$ and $M_{bm}^{(H)}$ into a new binary matrix, M_{bm}, to retain the results of both the horizontal and vertical scans. The connected components of M_{bm} are identified, considering a 8-neighbourhood, and only those containing more than one region are kept as crack region candidates; isolated crack region candidates are discarded (relabeled to '0'), as they are likely to correspond to oil spots or other types of noise.

Finally, the length of each retained connect component is computed and, for each image, the length of longest connected component (*llcc*) is stored. The selection of a given number of training images (controlled by the system operator) is achieved by sorting the entire image database in descending order of the computed *llcc* values – the TIS is chosen from the top of this sorted list. This procedure ensures that the images selected for training the classifiers effectively contain cracks.

Sample results of the binary matrices corresponding to images selected for the training step are shown in Fig. **4**, using k_1 and k_2 values equal to 0.4 and 2.0 respectively (empirically chosen by the system operator). More detailed results and the corresponding analysis are included in Section 6.1.

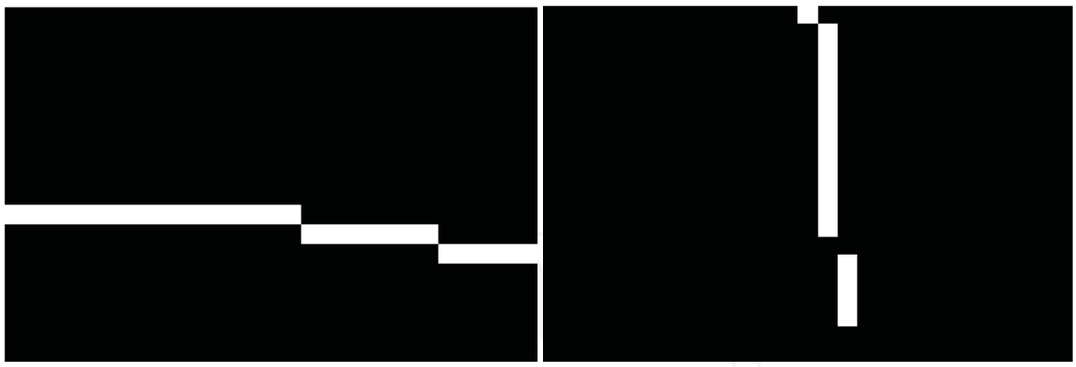


Fig. 4. Binary matrices showing the results of the preliminary crack region detection, for the right images of Fig. **2** and Fig. **3**, respectively. Regions in white are those preliminary classified as containing relevant crack pixels.

3.3 Image Normalization and Saturation

As stated in Section 3.1, pavement surface images were acquired during a survey over a Portuguese road using a digital camera. These images are free from shadows or other kind of occlusions, caused for instance by trees near road footpaths, but they present a non-uniform background illumination due to the type of sensor used, causing slight variations on the regions' pixel intensities average even in images without cracks.

To reduce this effect, an image normalization procedure is proposed. It consists in computing a base intensity level value (bil_{img}) for each image, equal to the average of the elements of M_{rm} corresponding to regions preliminary classified as not containing crack pixels, i.e., those labeled with value '0' in matrix M_{bm}. The need to use M_{bm} values for image normalization is the reason why this step is performed after the selection of training images.

Based on the bil_{img} value, a normalization constants matrix M_{nc} (with the same dimension of M_{rm}) is computed for each image, its elements being real values lower or higher than 1.0. The computation of M_{nc} elements is different depending if the corresponding label in M_{bm} is '0' or '1'.

For regions previously labeled with '0', i.e. regions preliminary classified as not containing cracks, the corresponding M_{nc} elements are computed using the expression in (6):

$$M_{nc}(\mathrm{i},\mathrm{j})^{'0'} = \frac{bil_{img}}{M_{rm}(\mathrm{i},\mathrm{j})^{'0'}} \tag{6}$$

where $M_{nc}(\mathrm{i,j})^{'0'}$ stands for the normalization constant to be applied to region (i,j), which has a M_{bm} label '0' and $M_{rm}(\mathrm{i,j})^{'0'}$ is the corresponding element in M_{rm}.

As an example, for a region with average pixel intensity of 163 and a M_{nc} value of 0.92, all that region's original pixel values are affected by this normalization constant. The resulting region average intensity will be 163×0.92=150.

For regions previously labelled with '1', i.e. regions preliminary classified as containing relevant cracks, the corresponding M_{nc} elements are computed using the expression in (7):

$$M_{nc}(\mathrm{i},\mathrm{j})'1' = \frac{bil_{img}}{\frac{1}{k^{(0)}}\sum_{p=-a}^{a}\sum_{q=-b}^{b} M_{rm}(\mathrm{i}+\mathrm{p},\mathrm{j}+\mathrm{q})'0'} \tag{7}$$

where $k^{(0)}$ is the number of regions with label '0' in a neighbourhood around the (i,j) region under analysis and the double sum accounts for all the corresponding M_{rm} elements. The search for regions with label '0' starts in 3×3 neighborhood (corresponding to a=b=1 in (7)). A larger neighborhood is adopted (e.g., 5×5 which corresponds to a=b=2 in (7)) only if no regions labeled '0' are found in the previous one. For instance, a region with label '1' and average pixel intensity of 152, with four neighbors labeled '0' and region averages of 148, 159, 140 and 153, has its original pixel intensities changed by a normalization constant of *152/150.*

Expression (7) only considers regions with label '0' for the computation of $M_{nc}(\mathrm{i,j})'1'$. This is done to prevent strong changes in pixel intensities of normalized regions with label '1', preventing dark pixels to become brighter than expected during the normalization step, thus avoiding to loose the information that this region is likely to contain a crack.

Sample results using the proposed normalization procedure are shown in Fig. **5**. The graph on the left shows M_{rm} original values, for the regions of the row considered in the right side of Fig. **3**; the graph on the right of Fig. **5** shows the normalized average intensity levels. As can be seen from Fig. **5**, the normalization procedure tends to equalize the average intensities for those regions preliminary classified as not containing cracks, while maintaining the average intensity of regions expected to contain crack pixels below bil_{img}.

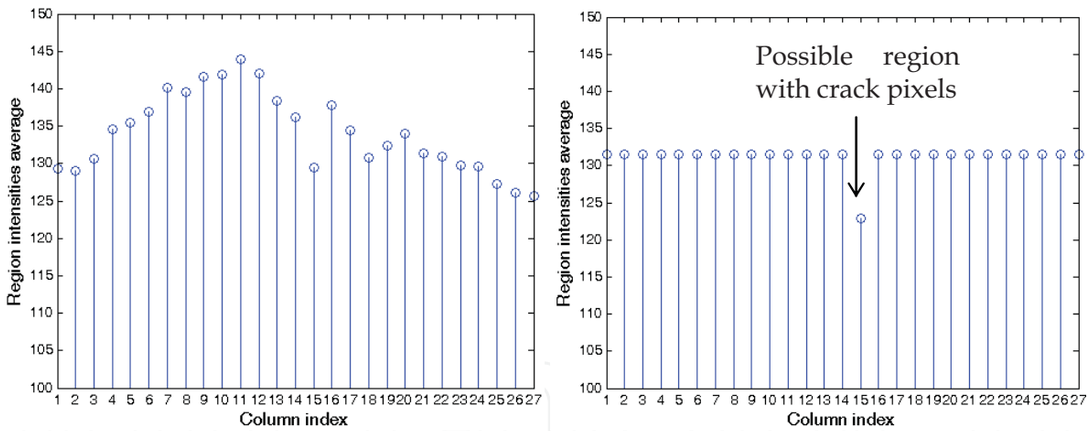


Fig. 5. Region average intensity values along the row selected in the right side of Fig. **3** before (left) and after (right) normalization.

Besides non-uniform background illumination, pavements surface images also frequently reveal the presence of white pixels due to specular reflectance of some surface materials. These pixels do not correspond to cracks but lead to higher intensity standard deviation values, even for regions without cracks. Higher standard deviation of region intensities are expected to be found in regions containing cracks (now due to higher differences between dark crack pixels and the corresponding average computed for the entire region). Therefore, white pixels may hinder detection performance, as different types of regions would present similar local statistics.

In order to eliminate the undesired influence of white pixels, a region saturation algorithm is proposed. For this purpose, the average of all pixel intensities of each normalized image is computed (*api*) and all image pixels having intensities higher than *api* assume that value. The pixel intensity saturation function is illustrated in Fig. **6**. The effect of applying the pixel intensity saturation algorithm to a normalized image is illustrated in Fig. **7**.

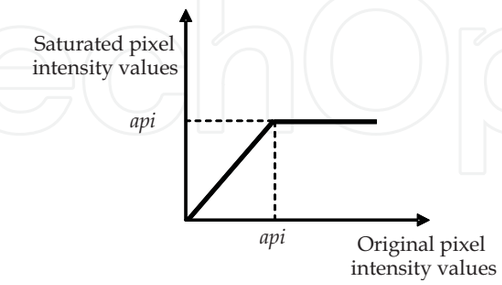


Fig. 6. Pixel intensity saturation function.

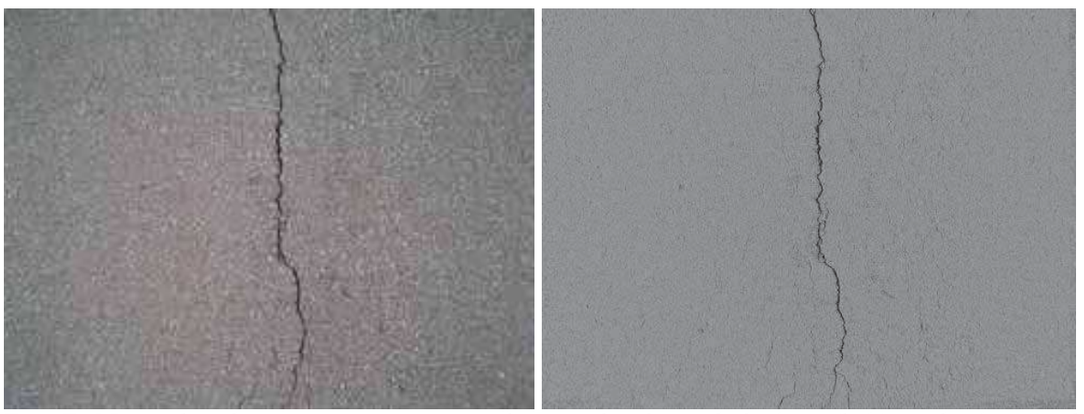


Fig. 7. Normalized image containing a longitudinal crack before (left) and after (right) applying the intensity saturation algorithm.

The proposed saturation function efficiently simplifies normalized images, reducing noise and also the standard deviation of regions without crack pixels, while keeping all relevant crack information.

To clarify the effect of applying the pixel saturation algorithm, which slightly changes the regions' average intensities, an example is shown in Fig. **8** for the row considered in the right image of Fig. **3**. At a first glance, comparing the right graph of Fig. **5** with the one on top of Fig. **8**, the region average intensities are globally lower for the second case. Moreover, the corresponding standard deviations are also lower after applying the saturation algorithm as seen in the bottom graphs of Fig. **8**. In fact, the average standard deviation value for the image regions preliminary classified as not containing cracks (26 out of the 27 regions in the example of Fig. **8**) is 26.8, while after applying the saturation algorithm it is reduced by approximately 54%, to 12.4. Still, for the region likely to contain cracks, the reduction is only 29% (31.5 against 44.1 in the non-saturated case).

Thus, the saturation algorithm achieves a strong standard deviation reduction for regions without cracks, creating a good separation to the standard deviation values of crack regions,

and allowing to consider it, together with the region average intensities, as the features to be exploited by the classifier used for crack regions detection, as discussed in the next section.

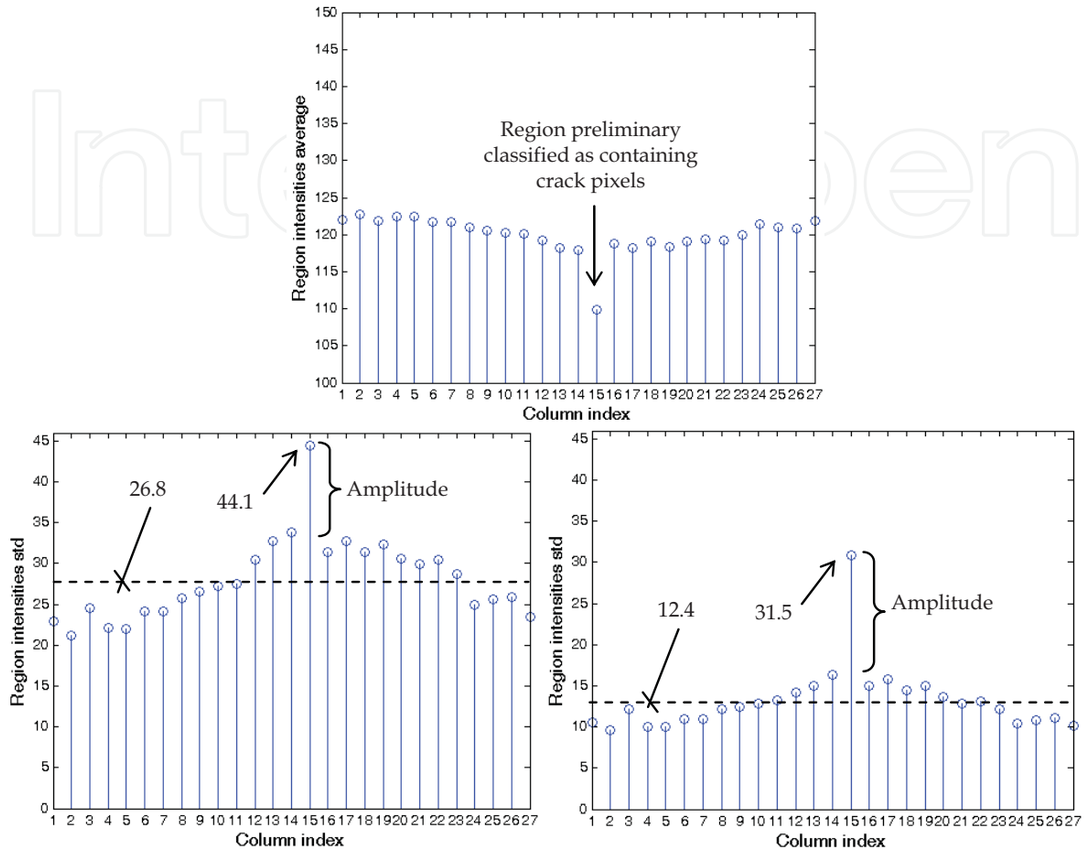


Fig. 8. Region average intensity values along the row selected in the right side of Fig. **3** after normalization and saturation (top) and standard deviation of region intensities for the normalized images before (bottom left) and after applying the saturation algorithm (bottom right).

3.4 Feature Extraction and Normalization

To automatically label regions as containing cracks or not, a pattern recognition system operating over a simple feature space is proposed. The feature space is two dimensional, being constructed using regions' local statistics, computed for normalized and saturated images. The first feature is the mean value of all pixel intensities in a region; the second is the standard deviation of the region's pixel intensities. Images can then be represented in the feature space - see example in Fig. **9**, where each point identifies a region of an image.

Since different images present different average values, as can be observed by the scattering of points in Fig. **9** top-right and bottom-left images, a further normalization step is needed to allow a better classifier performance.

This additional feature space normalization starts with the computation of each image's two dimensional feature space centroid, together with a global centroid computed for all the

database images. Then, for each individual image, the two dimensional feature space points are translated to align the respective centroid with the global one. The corresponding result is illustrated in the bottom-right image of Fig. **9**. Table 1 complements these results with the values of the intraclass and interclass distances (Heijden et al., 2004), computed for a TIS image set composed of five images, as discussed in Section 6.

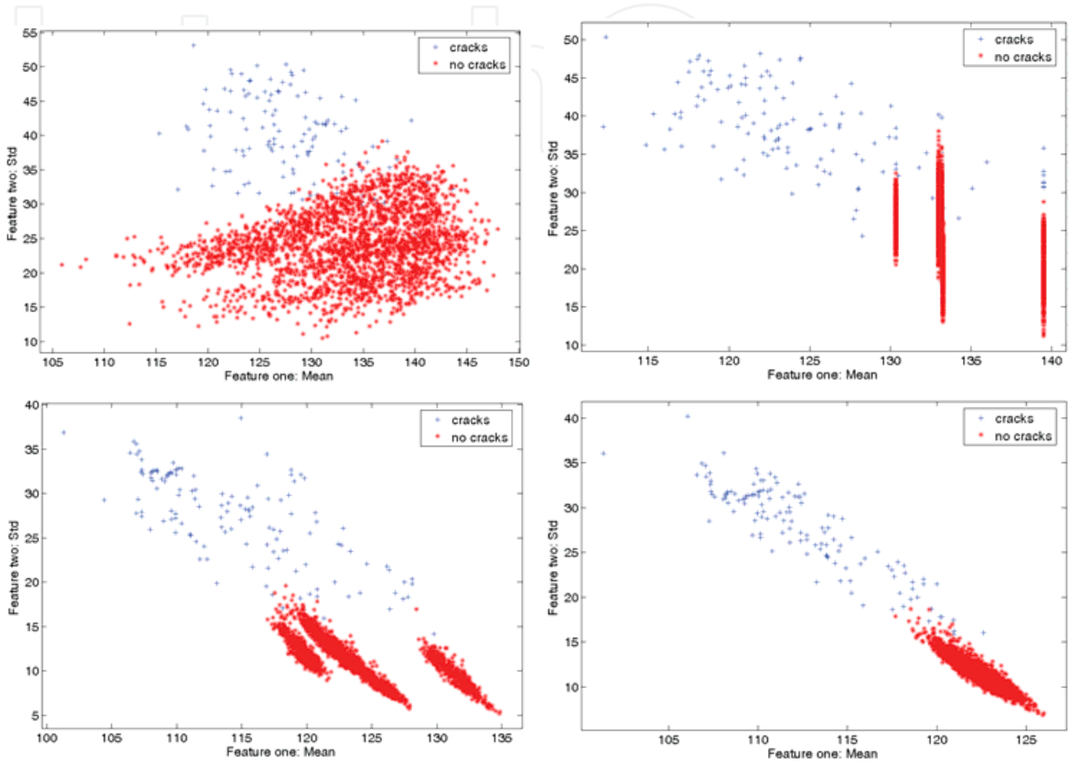


Fig. 9. Feature space representation, using a TIS composed of five images, for the original image (top-left), after image normalization (top-right), after normalization and saturation (bottom-left) and after the additional feature space normalization (bottom-right).

Implementations	**Intraclass distance** (crack regions)	**Intraclass distance** (no crack regions)	**Interclass distance**	**Crack region's intra/ interclass ratio** (%)	**No crack region's intra/interclass ratio** (%)
Original images	147.9	145.0	395.8	37.4	36.6
Norm.	150.4	59.1	371.4	40.5	15.9
Norm. + Satur.	138.7	45.5	423.9	32.7	10.7
Norm. + Satur. + Trans.	87.2	8.7	402.4	21.7	2.2

Table 1: Interclass and intraclass distances computed using TIS set.

As can be seen in the first line of Table 1, high intraclass and interclass distance values are obtained for the original images, denoting a very scattered feature space where class separation would be a difficult task, as illustrated by the top-right graph of Fig. 9.
After region normalization (top-right graph of Fig. **9**), non crack regions points become aligned along vertical lines (each vertical alignment corresponding to an image), with very little variation along the horizontal axis. For these points, the values of the second line of Table 1 show a better class compactness. The distribution of crack region's points is not significantly affected by this task.
Applying the saturation algorithm to the normalized images (see bottom-left graph in Fig. **9**) a reduction of the intraclass to interclass distance ratio is obtained for both classes.
With feature space normalization a further improvement is observed in the results. The intraclass to interclass distance ratios is the best (21.7% and 2.2%), revealing a more separable feature space and more compact point distributions.

4. Training and Classification

This section describes the classification strategies being evaluated, which are based on two supervised learning approaches: parametric (Section 4.1) and nonparametric (Section 4.2). Parametric approaches are based on a bivariate class-conditional normal density, as it provides a good data description (Oliveira & Correia, 2007).

4.1 Parametric Learning and Classification

Points obtained by applying the described feature extraction and normalization procedures to the training image set (TIS) are manually labeled by a skilled system operator, providing a training data set for which the labels are a priori known.
From a fully automatic application point-of-view this is a drawback, as a human operator is required to manually label image regions. However, since the aim here is to develop parametric supervised strategies for crack region detection, the manual labeling is required to create the training data to be used by the classifiers' parameter learning step.
All TIS feature points compose a pattern vector **x,** representing a sample of the random variable **X**, taking values on a sample space ***X***. For each element x_i of pattern vector **x,** one possible class y_i is assigned, where **Y** is the class set, .i.e. $y_i \in \mathbf{Y}$. Thus, the training set is:

$$\mathrm{T} = \left\{ (x_1, y_1) \ldots (x_n, y_n) : x_i \in \Re^2 ; y_i \in \{c_1, c_2\} \right\} \tag{8}$$

where n is the number of points of the pattern vector **x**. Only two classes are used: regions with crack pixels, labeled as class c_1, and regions without crack pixels, labeled as class c_2.
Assigning a loss penalty to misclassified measurements, the minimal expectation of the resulting cost is taken as an acceptable optimization criterion for the Bayesian classifier presented here (Heijden et al., 2004):

$$\hat{y}_i = \arg \max_{y_i} \ln \left(p(\mathbf{x} \mid y_i) p(y_i) \right) \tag{9}$$

where $p(y_i)$ are the class priors, computed by:

$$p(y_i = c_k) = \frac{\#\,\text{points labeled into class}\, c_k}{\text{total number of points for all classes}} . \tag{10}$$

with k being the class index. A loss function $L(s,a) : \mathbf{S}\times\mathbf{A} \rightarrow \mathbf{R}$ is constructed to quantify the cost of each classification action, where **S** is the state space, ***s*** is the true state of nature, **A** is the action space and a is the action (classification) taken by the classifier (Figueiredo, 2004). The decision rule is to take the action that minimizes the associated risk, i.e., take action a_1 if $R(a_1 \mid x)$ is lower than $R(a_2 \mid x)$, where a_k means classifying measurement x_i into class c_k with $k \in \{1,2\}$, symbolically represented by (Duda et al., 2004):

$$p(\mathbf{x} \mid y_i = c_1)P(y_i = c_1) \underset{c_2}{\overset{c_1}{\gtrless}} \left(\frac{L_{12} - L_{22}}{L_{21} - L_{11}}\right) p(\mathbf{x} \mid y_i = c_2)P(y_i = c_2) \tag{11}$$

where L_{pq} is the loss resulting from classifying a measurement into class c_p, while the true state of nature is class c_q, i.e. $L(\hat{y}_i = c_p \mid y_i = c_q)$. Since a uniform loss function is used here, i.e. $L_{11}= L_{22}=1$ and $L_{12}= L_{21}=0$, the expression in (10) identifies a *maximum a posteriori* probability classifier. Ground truth for the training set is known, thus the parameters for both classes are learned from TIS feature points, $X \sim N(\mu_k,\Sigma_k)$, with (Bishop, 2006):

$$\hat{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_{ki} \quad \text{and} \quad \hat{\Sigma}_k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k}(x_{ki} - \hat{\mu}_k)(x_{ki} - \hat{\mu}_k)^{\mathrm{T}} \tag{12}$$

where $\hat{\mu}_k$ is the sample unbiased vector mean, $\hat{\Sigma}_k$ is the sample unbiased covariance matrix, k is the class index and n_k is the total number of k class points.

Three ways to compute the decision boundaries are considered. The first one, denoted as linear, assumes a joint sample covariance matrix (Σ), with the boundary being computed by a weighted average (according to the class prior probabilities) of each class' covariance matrix, which results in a **linear decision boundary** (Duda et al., 2004; Heijden et al., 2004) given by:

$$\alpha + \mathbf{x}^{\mathrm{T}}\beta = 0 \tag{13}$$

$$\alpha = 2\ln\frac{P(y_i = c_2)}{P(y_i = c_1)} - \mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2 + \mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1 \tag{14}$$

$$\beta = 2\Sigma^{-1}(\mu_2 - \mu_1) \tag{15}$$

The second way to compute the decision boundary, denoted as quadratic, assumes a general covariance matrix resulting in the **quadratic boundary** (Heijden et al., 2004) defined by:

$$\alpha + \mathbf{x}^{\mathrm{T}}\beta + \mathbf{x}^{\mathrm{T}}\varphi\,\mathbf{x} = 0 \tag{16}$$

$$\alpha = +\ln|\Sigma_1| - \ln|\Sigma_2| + 2\ln\frac{P(y_i = c_2)}{P(y_i = c_1)} - \mu_2^{\mathrm{T}}\Sigma_2^{-1}\mu_2 + \mu_1^{\mathrm{T}}\Sigma_1^{-1}\mu_1 \tag{17}$$

$$\beta = 2(\Sigma_2^{-1}\mu_2 - \Sigma_1^{-1}\mu_1) \quad \text{and} \quad \varphi = -\Sigma_2^{-1} + \Sigma_1^{-1} \tag{18}$$

The third decision boundary, denoted as **independent**, is computed assuming independent features, i.e. the covariance matrices in (12) are now diagonal matrices computed as:

$$\hat{\Sigma}_{k_{l,l}} = \mathrm{E}\left[\left(\mathbf{x}_{k_l} - \mu_{k_l}\right)\left(\mathbf{x}_{k_l} - \mu_{k_l}\right)\right] \tag{19}$$

and $\hat{\Sigma}_{k_{l,m}}$ takes value zero whenever $l \neq m$; E stands for the expected value and l and m are feature identifiers, taking value 1 or 2 for class regions without or with crack pixels, respectively. Using these new covariance matrices, equations from (16) to (18) are used to compute the target decision boundary.

A sample result using the three types of decision boundaries, computed for the TIS, is illustrated in Fig. **10**.

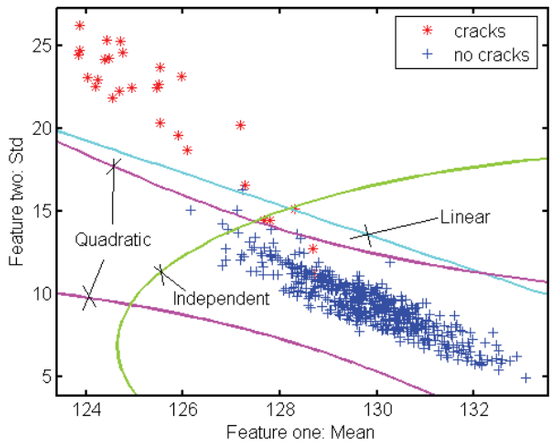


Fig. 10. Three parametric decision boundaries computed for the TIS.

4.2 Non-parametric Learning and Classification

This subsection deals with classifiers that operate when both conditional probability distributions are unavailable. This is different from the parametric case, where the only unknowns were the probability density parameters modeling the data.

In general, one advantage of non-parametric learning, when compared with parametric learning, is that not so much prior knowledge about the data to be processed is required, but, on the other hand, a large amount of data is needed to compensate the lack of knowledge about probability density functions, although it can be reduced when certain computational constrains of the classifiers apply (for example, the use of a linear boundary decision instead of a non-linear one) and they match the inherent distributions (Heihjen et al., 2004; Webb, 2002).

Here, three non-parametric techniques are considered: Parzen windows, k-Nearest Neighbor and Fisher's Least Square Linear classifiers.

The implemented **Parzen** algorithm for learning and classification follows the descriptions in (Heijden et al., 2004). Considering a labeled training vector **x** according to (8) and an unlabelled test set, the probability density estimation for an arbitrary test vector **z** is achieved by:

$$\hat{p}(\mathbf{z} \mid y_i = c_k) = \frac{1}{n_k} \sum_{q=1}^{n_k} \underbrace{\frac{1}{2\pi \times fs^2} \exp\left(-\frac{\|\mathbf{z} - \mathbf{x_q}\|^2}{\mathbf{2}fs^2} \right)}_{\mathbf{A}} \tag{20}$$

where **A** is a kernel that represents the knowledge about the distance between a test measurement **z** and the training measurement x_q, corresponding to a Gaussian interpolation distance function, n_k is the total number of measurements for class k and fs is a constant that controls the size of the kernel influence zone, computed such that it maximizes:

$$\sum_{k=1}^{2} \sum_{q=1}^{n_k} \ln\left(\hat{p}\left(\mathbf{x}_{k,q} \mid y_i = c_k \right) \right) \tag{21}$$

where $x_{k,q}$ is the sample q of the class k which is left out by the leave-one-out method when computing the estimation of the posterior probability density. A measurement is classified into class c_k with the maximum posterior probability:

$$\hat{k} = \underset{k=1,2}{\operatorname{argmax}} \left[\hat{p}(\mathbf{z} \mid y_i = c_k) \hat{P}(y_i = c_k) \right] \tag{22}$$

where $\hat{P}(y_i = c_k)$ represents class priors according to (10).

For **k-Nearest Neighbors** classification (*k-nn*), the estimated posterior probability density may have different resolutions when the training data is not homogeneous, i.e., it's resolution is higher when the training data is more dense. The posterior probability density for an arbitrary test vector **z** is computed by (Duda et al., 2001; Theodoridis & Foutroumbas, 2003):

$$\hat{p}(\mathbf{z} \mid y_i = c_k) \approx \frac{N_k}{n_k \mathbf{V}(\mathbf{z})} \tag{23}$$

where N_k is the number of samples inside the volume $\mathrm{V}(\mathbf{z})$—which represents a sphere centered in **z**—belonging to class k and n_k is the total number of training samples belonging to class k. Thus, a measurement is classified into the class (c_1 or c_2) that contains more training measurements in the N_k neighborhood of **z**:

$$\hat{k} = \arg\max_{k=1,2} \left\{ \hat{p}(\mathbf{z} \mid y_i = c_k) \hat{P}(y_i = c_k) \right\} = \arg\max_{k=1,2} \left\{ N_k \right\} \tag{24}$$

where $\hat{P}(y_i = c_k)$ again represents the class priors according to (10).

The aim of the **Fischer**'s linear classification strategy is to find the linear discriminant function between both classes, which corresponds to the projection that maximizes the class separability (Bishop, 2006; Duda et. al., 2001). Class separability in a direction $d \in \Re^n$ is defined by:

$$R(d) = \frac{d^T J_B d}{d^T J_W d} \tag{25}$$

which is also denoted as the ratio of the between-class covariance matrix (J_B) to the within-class covariance matrix (J_K), defined as:

$$J_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \tag{26}$$

$$J_W = J_{W_{c_1}} + J_{W_{c_2}} \quad , \quad J_{W_{c_k}} = \sum_{k=1}^{2} (\mathbf{x}_k - \mu_k)(\mathbf{x}_k - \mu_k)^T \tag{27}$$

where μ_k denotes the vector mean for class k, computed according to (12) and $\mathbf{x}_k$ is class k measurements vector data. An estimate of d is obtained maximizing (25) according to:

$$\hat{d} = \underset{d}{\mathbf{argmax}} \left(\frac{d^T J_B d}{d^T J_W d} \right) \tag{28}$$

Thus, a measurement from a vector $\mathbf{z}$ is classified into class c_1 when $y(x_i) \geq y_0$ for $y_0 = K.\mathbf{z}$ ($\mathbf{z}$ is classified into class c_2 otherwise).
A sample result using the three types of decision boundaries, computed using the TIS, is illustrated in Fig. **11**.

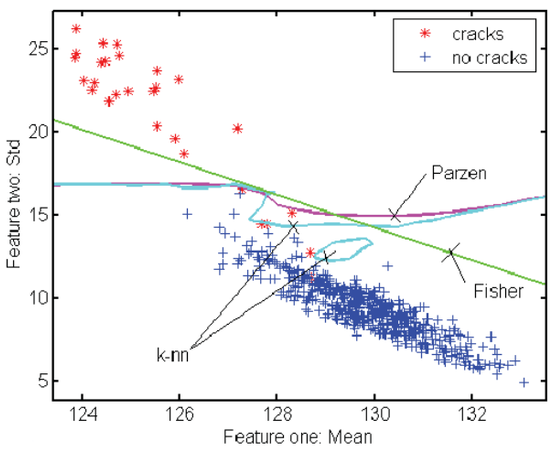


Fig. 11. Three non-parametric decision boundaries computed for the TIS. For *k-nn*, the boundary shown corresponds to a neighborhood of 1 point.

5. Crack Type Classification

Detection results are stored in binary matrices (one for each TTIS image) with the same dimensions as (1), where '1' means regions labeled as containing crack pixels and '0' the opposite case. All binary matrices are then processed to identify connect components and the resulting connected crack regions are finally classified into one of the crack types considered in the scope of this research work, following the specifications of the Portuguese Distress Catalog (JAE, 1997): longitudinal (c_L), transversal (c_T) or miscellaneous (c_M).
Crack type classification uses another pattern classification system exploiting a new 2D feature space. A crack type label is assigned to each connected crack region and cumulatively added to each TTIS image.
The 2D feature space used for crack type classification is composed by the standard deviations of the column (feature one) and row (feature two) coordinates of connected crack regions. A sample representation of this feature space is given in Fig. **12**.

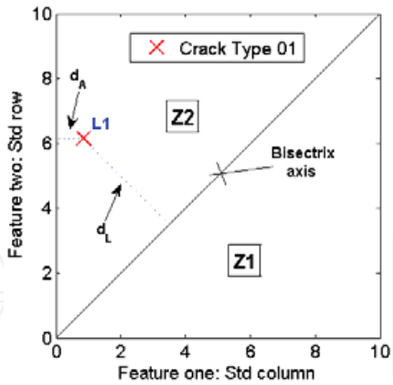


Fig. 12. 2D feature space used for crack type classification. Point L_1 represents a connected crack region classified as a 'longitudinal crack'.

The bisectrix sectioning the 2D feature space into two zones, 'Z1' and 'Z2', represents the points where connected components have equal column and row standard deviation values, identifying perfect miscellaneous cracks. Points positioned over the horizontal or vertical axes correspond to perfect transversal or longitudinal cracks, respectively.
Crack type classification is performed by computing two distances for each connected crack region point representation in the 2D feature space: d_L and d_A, where d_L is the distance from the point to the bisectrix axis and d_A corresponds to the distance to nearest axis (horizontal or vertical). The example in Fig. **12** shows the classification of one connected crack region (point L_1) as a 'longitudinal crack' ($d_L > d_A$). This crack type classification is fully automatic and unsupervised, no training stage being required.
The probability of a crack belonging to class c_L or c_T is computed, according to:

$$P(y_i = c_{cr} \mid r_i) = 1 - \frac{d_{Ai}}{d_{Ai} + d_{Li}} \tag{29}$$

while the probability of a crack belonging to the miscellaneous cracks class (c_M) is computed according to:

$$P(y_i = c_M \mid r_i) = 1 - \frac{d_{Li}}{d_{Ai} + d_{Li}} \tag{30}$$

where the index $_{cr}$ is one of the class indexes T or L, d_{Ai} is the distance from point i to the nearest axis, d_{Li} is the distance from point i to the bisectrix and r_i is the observation (region i).
Thus, a connected crack region is classified into the class presenting a probability above 0.5:

- a crack is classified as 'longitudinal' (class c_L) if $d_L > d_A$ and the nearest axis is the vertical one;
- a crack is classified as 'transversal' (class c_T) if $d_L > d_A$ and the nearest axis is the horizontal one;
- a crack is classified as 'miscellaneous' (class c_M) if $d_A > d_L$, independently of the nearest axis.

6. Experimental Results and Performance Evaluation

The proposed classification strategies are evaluated over the TTIS, which is composed by real flexible pavement surface images, eventually containing cracks with linear development. These images were acquired during a survey over a Portuguese road and ground truth data has been manually constructed. Part of the algorithmic development was supported by the PRtools toolbox (Duin et al., 2004). Experimental results are firstly presented for crack regions detection (Section 6.1) and then for crack type classification (Section 6.2).

6.1 Crack Regions Detection Results and Evaluation

Sample results for one TTIS image using the available classifiers are shown in Fig. **13**. For the k-nn strategy, one nearest neighbor (*1-nn*) is considered, as this is the neighborhood that optimizes the leave-one-out error for the target image.

An evaluation of the different strategies, by comparison with the ground truth data, is included in Table 2. A global Error-rate is computed (e-r_G being the classification error for classes c_1 and c_2), as well as some metrics related only to regions with crack pixels: Crack Error-rate (e-r_{Cr}), Precision (pr), Recall (re) as well as a Performance Criterion (pc) reflecting the overall classifier performance, according to (Tax, 2006):

$$e - r_G = \frac{\text{Number of regions wrongly classified for classes } c_1 \text{ and } c_2}{\text{Total number of regions}} \tag{31}$$

$$e - r_{Cr} = \frac{\text{Number of regions wrongly classified for class } c_1}{\text{Total number of crack regions (ground truth)}} = 1 - re \tag{32}$$

$$pr = \frac{\text{Number of regions correctly classified for class } c_1}{\text{Total number of crack regions detected}} \tag{33}$$

$$re = \frac{\text{Number of regions correctly classified for class } c_1}{\text{Total number of crack regions (ground truth)}} \tag{34}$$

$$pc = \frac{2 \times pr \times re}{pr + re}. \tag{35}$$

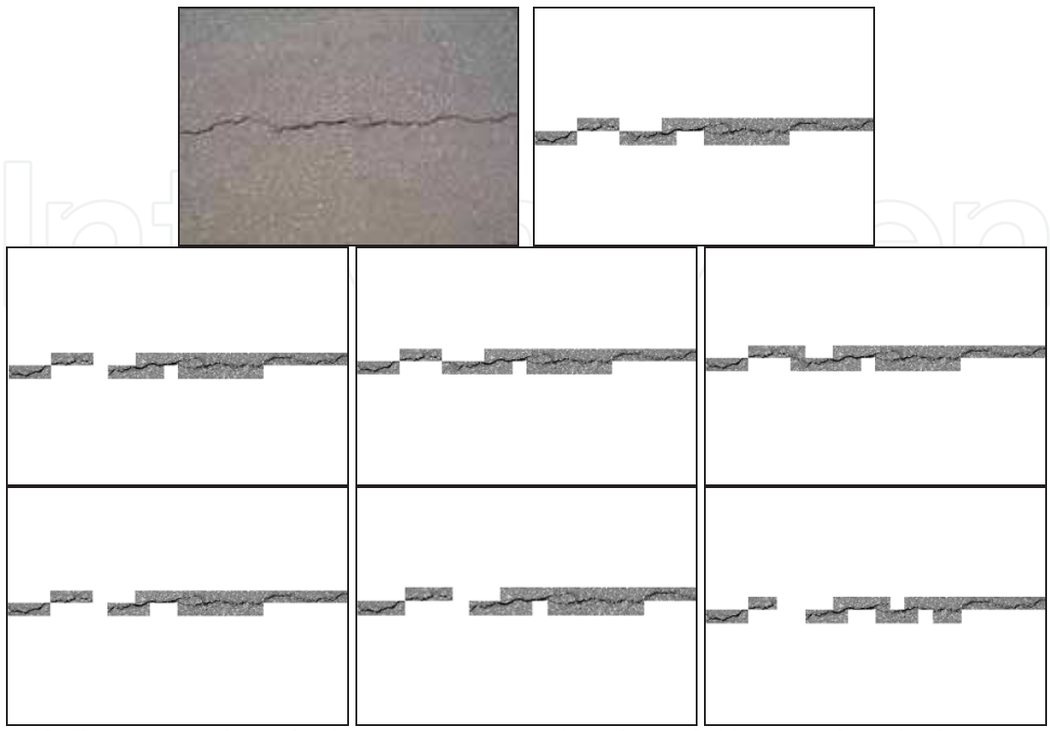


Fig. 13. Experimental results for a test image: original (top left), ground truth classification (top right). Parametric classification results (2nd line): linear classifier (left), quadratic classifier (middle), classifier with independent features (right). Non-parametric results (3rd line): Parzen windows (left), 1-nn nearest neighbors (middle) and Fischer's linear classifier (right).

Strategy	**Global Error-rate** ($e\text{-}r_G$)	**Crack Error-rate** ($e\text{-}r_{Cr}$)	**Precision** (pr)	**Recall** (re)	***pc***
Linear	0.68%	8.90%	97.1%	91.2%	94.0%
Quadratic	**0.64%**	**2.96%**	92.5%	**97.0%**	**94.7%**
Independ.	0.85%	6.87%	92.4%	93.1%	92.7%
Parzen	0.73%	10.79%	**98.2%**	89.2%	93.3%
k-nn	0.78%	5.38%	92.5%	94.6%	93.5%
Fischer	1.00%	15.45%	98.1%	84.6%	90.7%

Table 2. Detection results for regions with crack pixels case. Best results for each metric are underlined.

The best overall classifier performance is achieved by the quadratic classifier, according to *pc* values and confirmed by the best Recall value, meaning that this classifier produces the best true positive detection performance.

An interesting observation is that the features used seem to have some degree of dependence, which can be seen by comparing the quadratic and the independent parametric classifier results, but a worst classification performance is achieved when a diagonal covariance matrix is assumed. The use of parametric classifiers seems to be a good strategy,

producing better *pc* values and taking into account that Recall is more important than Precision for this type of application.

It is important to note that although the use of *k-nn* classifier produces good results (see *pc* and Recall), it may be difficult to obtain a fixed neighborhood size. For different training images, values between 1 and 10 were observed as the best, with an average of 4. Using a small neighborhood may produce some over fitting problems, with the decision boundary adapted to the training set, thus leading to a poor generalization of the classifier performance.

Additionally, all classifiers seem to perform very well according to false positives detection (i.e., regions without crack pixels being classified as containing cracks), with the corresponding computed errors always below 1%.

Looking in more detail to the quadratic classifier results, some samples computed for TTIS images and the respective ground truths are shown in Fig. **14**, emphasizing the good performance of the classifier.

It is also interesting to compare these results with those obtained in the preliminary classification stage for selecting images for the TIS (see Section 3.2). The corresponding results for the same metrics reported in Table 2 are included in Table 3.

Comparing the values reported in Table 2 and Table 3, it can be noticed that at the preliminary classification strategy achieves very good precision results (95.7%). This means that the great majority of crack regions preliminary detected do correspond to image regions containing crack pixels, which is important at that stage as it effectively finds good images for the training set.

Apart from that, crack detection using a Normal based density quadratic classifier significantly raises the system performance (from 66.1% to 97.0% for recall), although more false positives are detected in this case (precision drops from 95.7% to 92.5%).

Global Error-rate ($e\text{-}r_G$)	**Crack Error-rate** ($e\text{-}r_{Cr}$)	**Precision** (pr)	**Recall** (re)	***pc***
1.7%	33.9%	95.7%	66.1%	76.7%

Table 3. Results for crack regions preliminary classification.

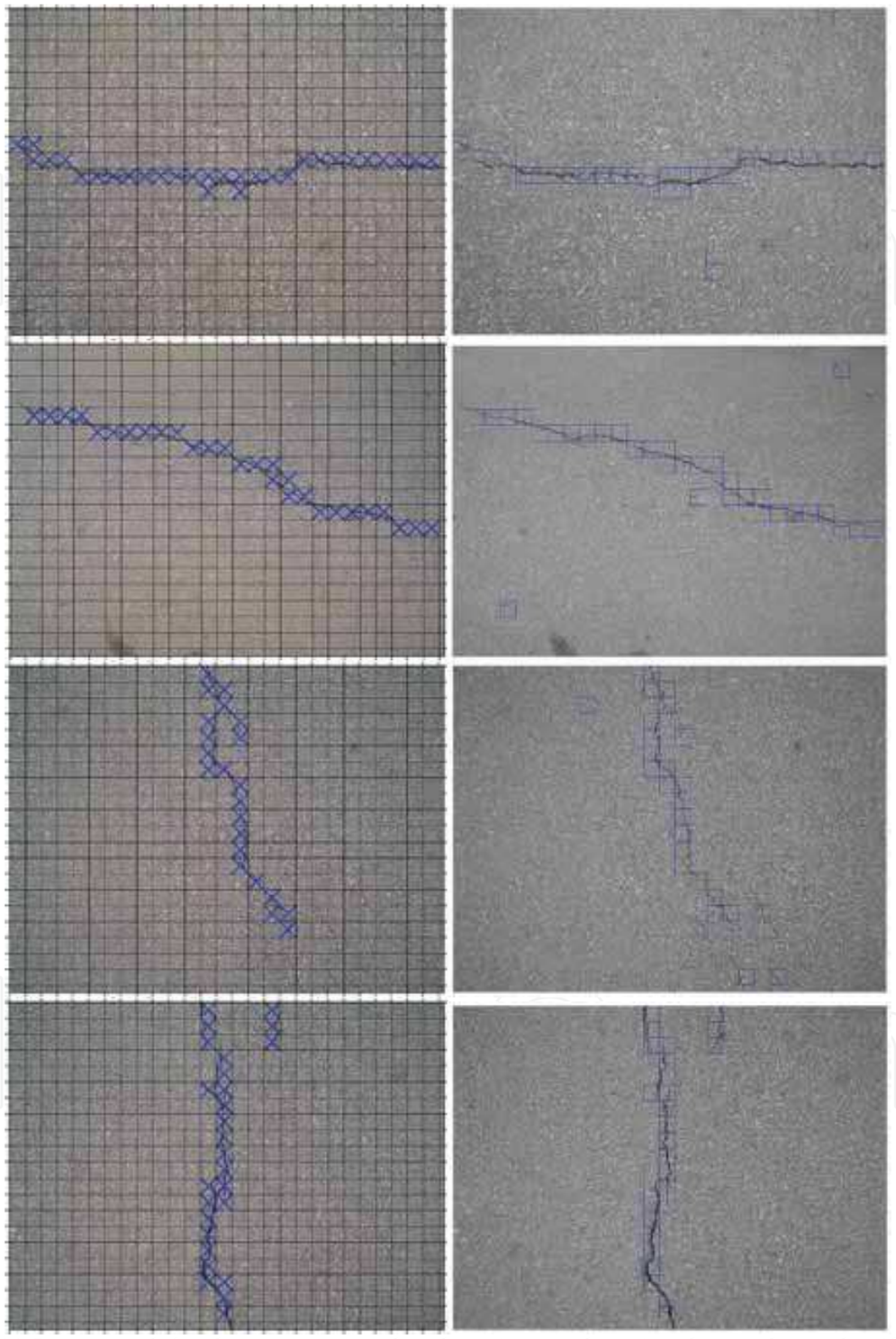


Fig. 14. Images on left column correspond to detection results using the quadratic classifier. The right column includes the corresponding ground truth.

Fig. 15 shows ground truth, preliminary classification and crack detection results (first, second and third column respectively) for the same images presented in Fig. **14**.

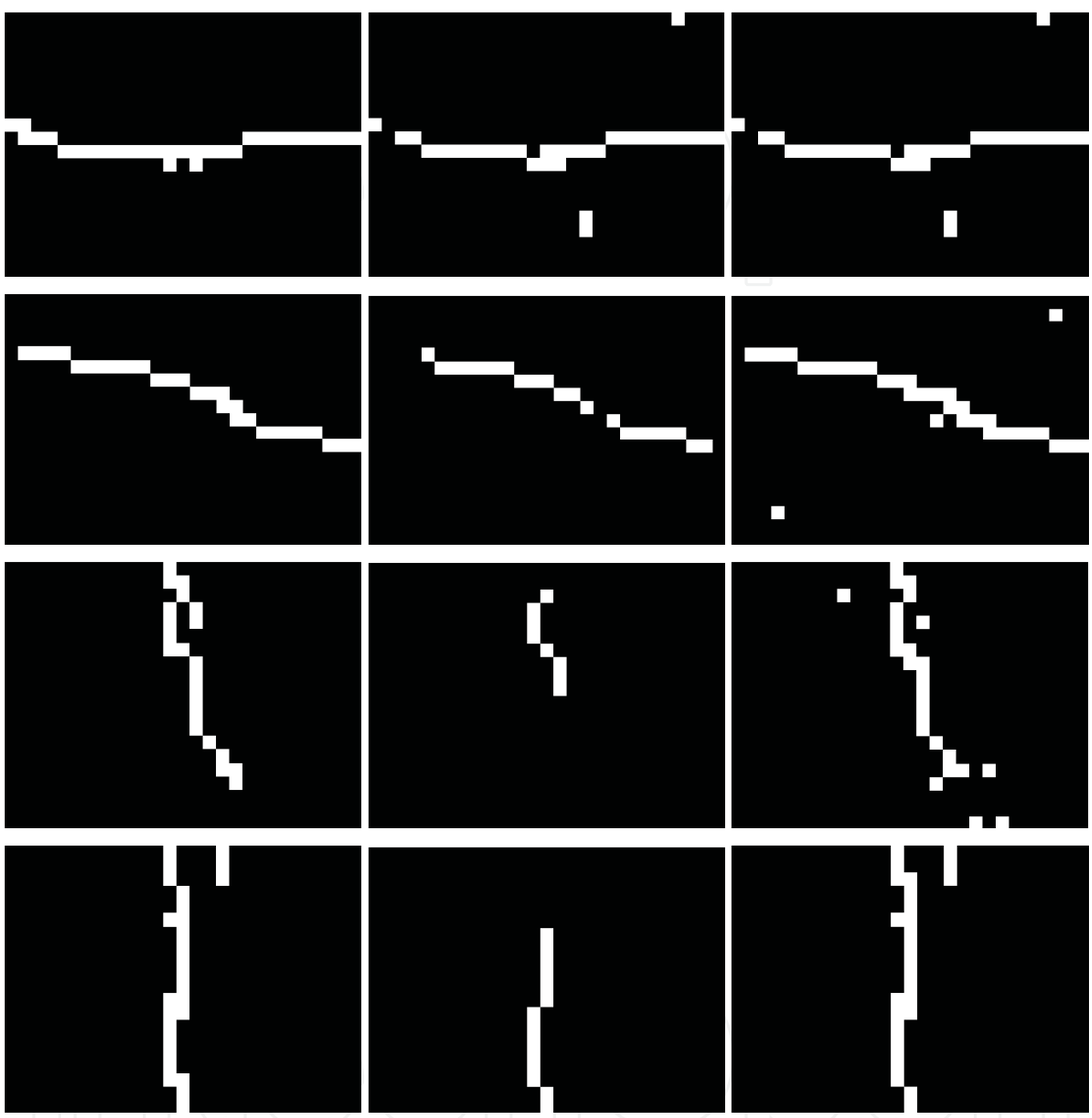


Fig. 15. Results of the preliminary crack regions detection (left), crack detection using a quadratic classifier (middle) and the corresponding ground truth (right).

6.2 Crack Type Classification Results and Evaluation

Crack type classification is performed on the resulting binary images produced by the crack detection task. Crack type classification labels are used to annotate database images and can later be used by a search engine to retrieve images containing a given type of crack. Fig. **16** shows crack classification results for the sample images shown in Fig. **14**.

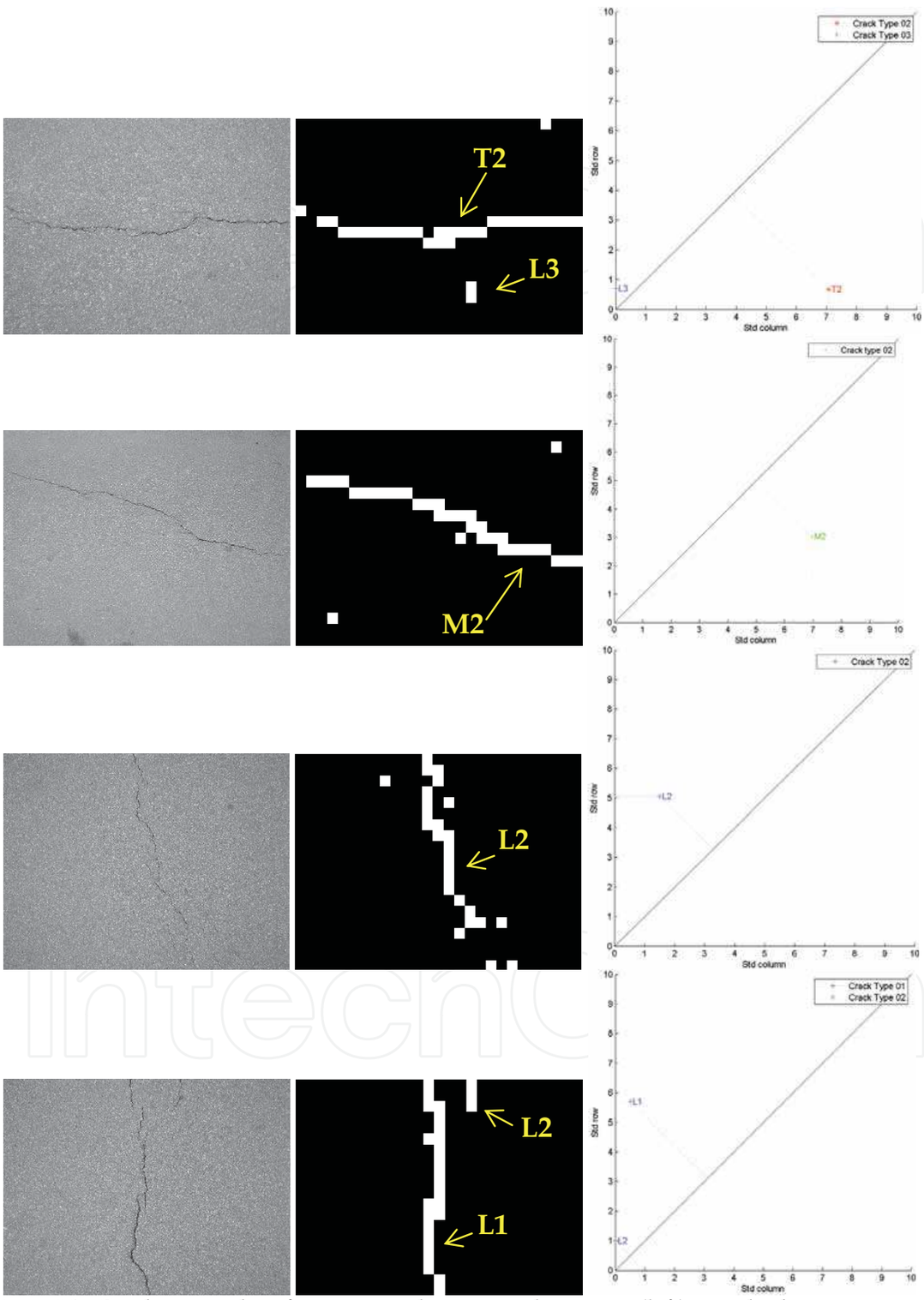


Fig. 16. Crack type classification results: original images (left), crack detection results (middle) and the corresponding crack type classification feature space (right).

From top to bottom, the first line shows an image containing a transversal crack and a short longitudinal crack. The second contains a miscellaneous crack. The third and fourth lines show images containing longitudinal cracks. The column on the right shows the crack's representation in the 2D feature space used for crack type classification. Regions with length equal to '1' (isolated regions) are not considered in the classification process, as they are likely to correspond to oil spots or similar occurrences in pavement surface images.
Using the crack type classification ground truth constructed for the TTIS, 100% recall and precision are obtained for all the cracks, emphasizing a very good classifier performance.

7. Conclusions and Future Work

This chapter proposes a supervised system for crack regions detection and classification. The proposed system automates the selection of training images, splitting the image database into training and test sets.
Six supervised classification strategies (three parametric and three non-parametric) were tested and analyzed. All six obtain an acceptable performance, with parametric classifiers, and especially the quadratic one, achieving the best classification results.
All detected cracks were correctly classified into types, considering a set of crack types listed in the Portuguese Distress Catalogue (JAE, 1997).
In terms of future developments, filtering techniques may be introduced to preprocess images before the classification stage, notably for reducing falloff and specular reflection problems. Also unsupervised approaches may be developed and confronted with those presented in this chapter, notably investigating the use of one class classifiers, as regions with crack pixels may be seen it as outliers of a well defined cluster of points in the feature space.
Additionally, a reject-option for the Bayesian approach and a non uniform loss function will be explored, since false positive detections have less impact than false negatives detection. Also a deeper study of windows size, to maximize class separability, will be performed.

8. References

Bishop, C. (2006). Pattern Recognition and Machine Learning, Springer, ISBN: 0-387-31073-8, USA

Chambon, S., Subirats, P. & Dumoulin, J. (2009). Introduction of a wavelet transform based on 2D matched filter in a Markov random field for fine structure extraction: application on road crack detection, in Proceedings of IS&T/SPIE Electronic Imaging, - Image Processing: Machine Vision Applications II, San José, USA

Chen, H. & Miyojim, M. (1998). Automatic pavement distress detection system. Journal of Information Sciences, Vol., 108, (July 1998) pp. 219-240

Chou, J., O'Neill, W. & Cheng, H.D. (1994), Pavement distress classification using neural networks, IEEE International Conference on Systems, Man, and Cybernetics - "Humans, Information and Technology", pp. 397 – 401, October 1994

Duda, R., Hart, P. & Stork, D. (2001). Pattern Classification, John Wiley & Sons Ltd, ISBN: 0-471-70350-8, Canada

Duin, R., Juszczak, P., Paclik, P., Pekalska, P., Ridder, D. & Tax, D. (2004). PRTools 4 - A MatLab Toolbox for Pattern Recognition – version 4.0.23, http://www.prtools.org/, Netherlands

Figueiredo, M. (2004). Lecture Notes on Bayesian Estimation and Classification, http://www.lx.it.pt/~mtf/learning/, Portugal

Heijden, F., Van der, Dwin, R.P.W., Ridder, D. & Tax, D.M.J. (2004). Classification, Parameter Estimation and State Estimation: An Engineering Approach using Matlab, John Wiley & Sons Ltd, ISBN: 0-470-09013-8

Huang, Y. & Xu, B. (2006). Automatic inspection of pavement cracking distress, Journal of Electronic Imaging, Vol. 15, N.1, (Jan-Mar 2006), SPIE and IS&T

JAE. (1997). Catálogo de Degradações dos Pavimentos Rodoviários Flexíveis – 2ª Versão, Ex-Junta Autónoma das Estradas, Portugal

Li, L., Chan, P., Rao, A. & Lytton, R.L. (1991). Flexible pavement distress evaluation usin image analysis, in Proceedings of the Second International Conference on Applications of Advanced Technologies in Transportation Engineering, pp. 66-70, 18-21 August

Liu, F., G. Xu, Yang, Y., Niu, X. & Pan, Y. (2008). Novel approach to pavement cracking automatic detection based on segment extending, in International Symposium on Knowledge Acquisition and Modeling. KAM '08, pp. 610-614, 21-22 December

Ma, C., Zhao, C. & Hou, Y. (2008). Pavement distress detection based on nonsubsampled contourlet transform, in International Conference on Computer Science and Software Engineering, pp. 28-31, 12-14 December

Meignen, D., Bernadet, M. & Briand, H. (1997). One application of neural networks for detection of defects using video databases: identification of road distresses, Proceedings of 8th Int. Workshop on Database and Expert Systems Applications, pp. 459-464, 1-2 September 1997

Oliveira, H. & Correia, P.L. (2007). Automatic crack pavement detection using a Bayesian stochastic pattern recognition system, Proceedings of RECPAD2007, Portugal, October 2007, Lisbon

Oliveira, H. & Correia, P.L. (2008). Supervised strategies for crack detection in images of road pavement flexible surfaces, Proceedings of EUSIPCO2008, Switzerland, August 2008, Lausanne

Qingquan, L. & Xianglong, L. (2008). Novel approach to pavement image segmentation based on neighboring difference histogram method, in Congress on Image and Signal Processing CISP '08, pp. 792 – 796, Volume 2, 27-30 May 2008

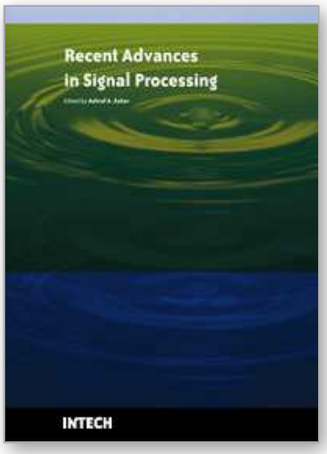
Tax., D. (2006). Data Description Toolbox, http://ict.ewi.tudelft.nl/~davidt/dd_tools.html, Netherlands

Theodoridis, S. & Foutroumbas., K., (2003). Pattern Recognition – 3nd edition, Elsevier – Academic Press, ISBN: 0-123-69531-7, USA

Wang, K.C.P. (2000). Designs and implementations of automated systems for pavement surface distress survey, Journal of Infrastructure Systems, Vol. 6, N.1, (March 2000), ASCE, ISSN 1076-0342/00/0001-0024–0032

Webb, A. (2002). Statistical Pattern Recognition – 2nd edition, John Wiley & Sons, ISBN: 0-470-84514-7, England

Zhang, H. G. & Wang Q. (2004). Use of artificial living system for pavement distress survey, The 30th Annual Conference of IEEE Industrial Electronics Society, Korea, November 2004, Busan



Recent Advances in Signal Processing

Edited by Ashraf A Zaher

ISBN 978-953-307-002-5

Hard cover, 544 pages

Publisher InTech

Published online 01, November, 2009

Published in print edition November, 2009

The signal processing task is a very critical issue in the majority of new technological inventions and challenges in a variety of applications in both science and engineering fields. Classical signal processing techniques have largely worked with mathematical models that are linear, local, stationary, and Gaussian. They have always favored closed-form tractability over real-world accuracy. These constraints were imposed by the lack of powerful computing tools. During the last few decades, signal processing theories, developments, and applications have matured rapidly and now include tools from many areas of mathematics, computer science, physics, and engineering. This book is targeted primarily toward both students and researchers who want to be exposed to a wide variety of signal processing techniques and algorithms. It includes 27 chapters that can be categorized into five different areas depending on the application at hand. These five categories are ordered to address image processing, speech processing, communication systems, time-series analysis, and educational packages respectively. The book has the advantage of providing a collection of applications that are completely independent and self-contained; thus, the interested reader can choose any chapter and skip to another without losing continuity.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen