One Application of Neural Networks for Detection of Defects Using Video Data Bases: Identification of Road Distresses

D. Meignen (1) M. Bernadet (2) H. Briand (2)

(1) Laboratoire Central des Ponts et Chaussées (L.C.P.C.) - Centre de Nantes - B.P. 19 44340 Bouguenais - FRANCE e-mail: daniel.meignen@lcpc.fr

(2) IRIN University of Nantes 2, rue de la Houssiniere - BP 92208 44322 Nantes cedex 3 - FRANCE e-mail: {mbernade, hbriand}@ireste.fr

Abstract

We describe here one application of neural networks to the discovery of road distresses from video sequences. After describing the context of this application, we detail its progressive design: we initially thought of using only one neural network to analyze totally each image extracted from a video sequence, we later thought of a more simple neural network analyzing each time only a small part of each image, with a preprocessor scanning the image; we finally preferred to simplify the role of the neural network by putting ahead one image pre-processing sequence to extract objects identified then by the neural network. We describe the sequence of treatments that we use and detail each processing step: improvement of the original image, extraction of significant objects (possible distresses) and identification of these objects by the neural network. We conclude by evaluating performances of our system and by discussing possible improvements.

1. Introduction

This paper describes the recent developments of a study which is in progress at the Laboratoire Central des Ponts et Chaussées (L.C.P.C.), in Nantes (France). The L.C.P.C. is a French public laboratory who has in charge the research and development of new methods for the conception, construction, preservation and repair of roads and bridges. Within the tasks assigned to this laboratory, the management of pavement maintenance has a main importance. This task requires a lot of data to describe the nature and condition of pavements, and among these data, the surface distresses are of major interest. Up to now, these informations were collected mainly through visual surveys. In France, a special system, named DESY, was developed and has been used for more than ten years. Unfortunately, this system is slow (< 10 km/h) and is dangerous for both the operators and the roads users. Furthermore, the results can differ from an operator to another for the same road, specially when expressing the gravity of the distresses. In order to overcome these difficulties, researches have been intensively conducted for several years (for instance [1], [4], [7], [8]). The L.C.P.C. wishes then to automate and fasten the process of surveying road pavements with detection and characterization of road surface distresses.

For this purpose, we have realized a system which analyzes sequences of video images to discover among them possible pavements distresses. The design of this system has considered successively three architectures.

2. The design of our system

At first, we considered the possibility to use directly a neural network with its inputs connected to all the pixels of one image in the video sequence. This solution appeared as the simplest in the possible architectures, since it needed only one neural network.

However, it appeared soon that this solution required a neural network with as many inputs as the number of pixels of one image and as many outputs as the number of these inputs, but with only three possible values for the outputs: presence of a distress, absence of a distress, or doubt about the presence of a distress on one point.

So, the complexity of this network should have induced prohibitive learning times, and probably a long exploitation time. A second potential solution was then examined: from the remark that the identification of pixels associated to road distresses could essentially be done by relating these pixels to their neighbors, we imagined to reduce the size of the neural network by limiting the scope of its consideration to a small window inside the image, and by scanning the image with this window. It appeared then that if the learning time would probably have been reduced to reasonable values, the analysis time risked to be too strongly increased.

We then decided to try an completely different method by putting ahead of the neural network a preprocessing sequence of treatments to extract significant objects from the image and by using the neural network to identify these objects. It is this solution that we have realized, it gives rather good identification rates and its learning times are correct, and if the processing times are a little too slow; this problem can be easily solved by using a quicker processor or by using a specific hardware system in the last phase of the application.

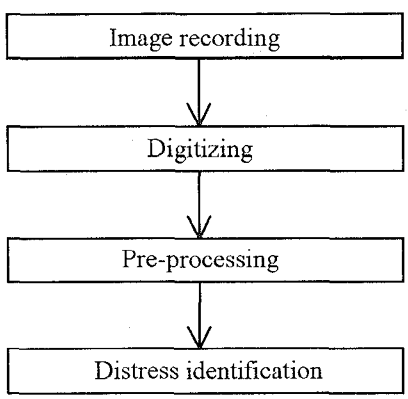

Figure 1. Structure of the system

The complete system is described by figure 1. It is based on an analysis of sequences of video images of the road. A video camera is installed outside a van (vertically ahead of the van). Images of the road are continually recorded on video tapes. Each view represents about 3.2m*4 m of pavement surface. The images are digitized to 640 x 442 pixel digital images. Each pixel represents a gray level between 0 and 255, 0 indicating a dark pixel and 255 a white pixel. The next step consists in the creation of a binary image containing only dark (distressed) and white (non distressed) pixels, by using a threshold. Finally, this image is operated by a neural network to identify the nature of the distress.

3. The pre-processing phase

The pre-processing phase can be divided in two steps. The first one consists in improving the initial image in order to have more contrast between distressed pixels and non distressed pixels. In the second step, distressed pixels are extracted from the improved image to obtain the binary image.

3.1 Improvement of the original image

Before creating the binary image, several techniques are applied to improve the original image, inspired from [2], [5], [10]. Figure 2 presents an image and its histogram. One can see that most pixels are located in a small portion of the histogram: there is not enough contrast, so the extraction of distressed pixels is difficult. Therefore, the first step of the pre-processing extends the histogram to obtain a larger distribution of gray levels; this is realized by the following method:

- two points x1 and x2 are selected on the histogram,
- all pixels with a gray level less than x1 are put to 0,
- all pixels with a level higher than x2 are put to 255,
- all pixels between x1 and x2 are modified by:

$$Gray\_level_{new} = a * Gray\_level_{old} + b$$

with $a = \dfrac{255}{x2 - x1}$ and $b = -\dfrac{255 - x1}{x2 - x1}$.

The positions of x1 and x2 are determined from the position x of the maximum of the histogram h(x): x1 and x2 are the positions on the left and the right of x such as h(x1) and h(x2) are lower than 1% of h(x). Starting from the original image of figure 2, this method and its result are depicted by figure 3. On the resulting image, the distress (a transversal crack) appears more clearly.

Another way to increase the contrast in an image is to use convolutions. A convolution is a weighted summation process. Each pixel in the neighborhood of one pixel (as shown on figure 4) is multiplied by a similarly dimensioned convolution kernel, and the resulting sum replaces the value of the central pixel. Each element of the convolution kernel is a weighting factor. The choice of the weightings factors determines the type of transformation: convolutions can effect image sharpening, image softening, edge enhancement.

Convolutions can also be used to reduce the noise in an image, with a different algorithm. The pixels in the neighborhood are sorted (figure 5) and the result is the median pixel value of the sorted vector. Our application uses only one convolution named "North-East" for image edge enhancement.

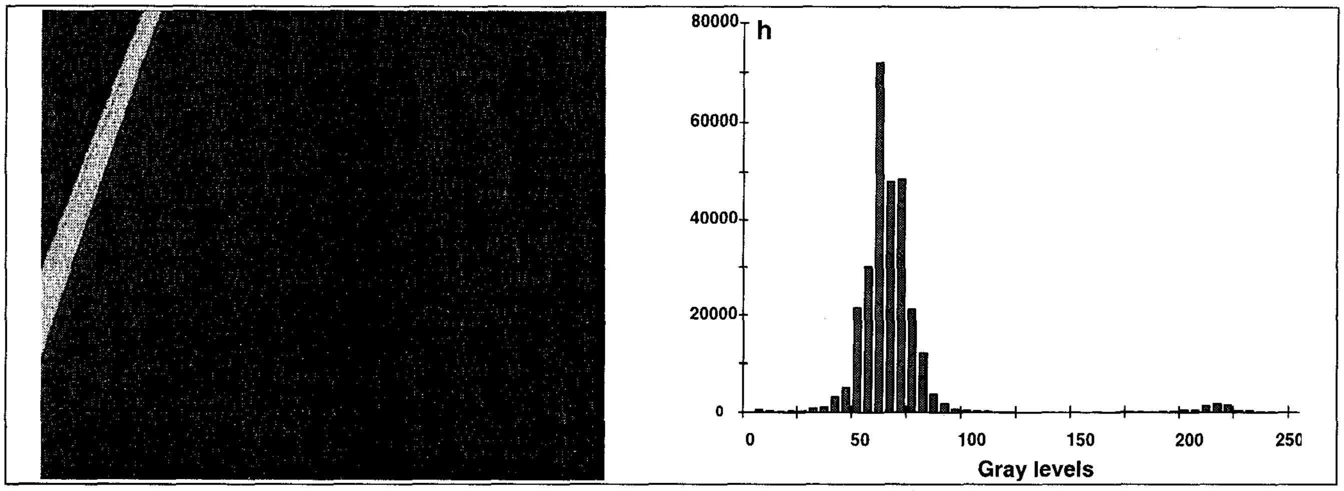

Figure 2. One image and its gray level histogram

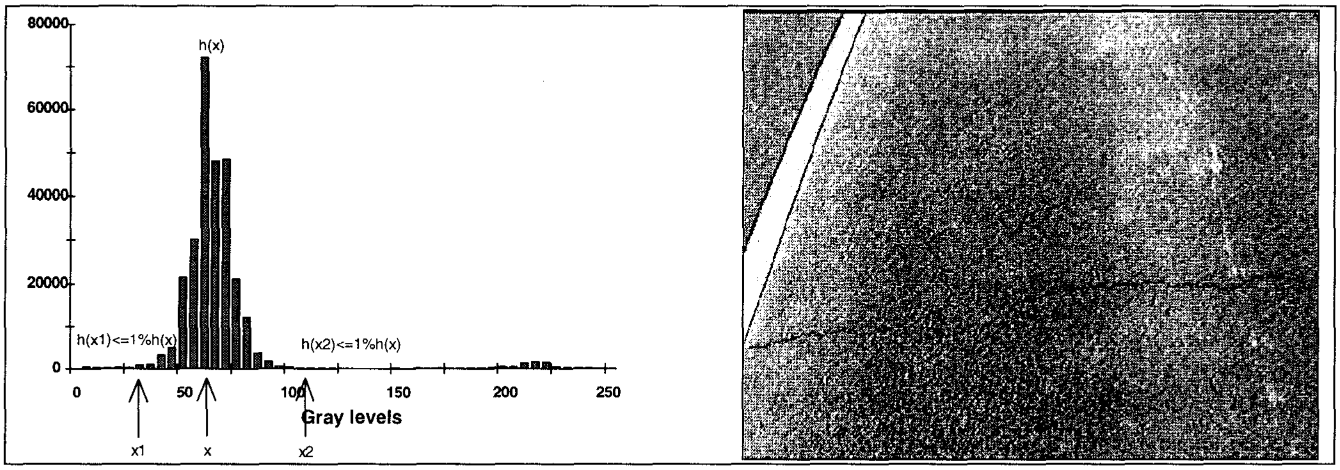

Figure 3. Result of the extension

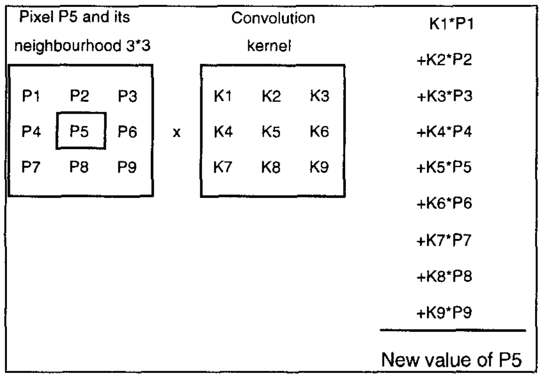

Figure 4. Principle of the convolution

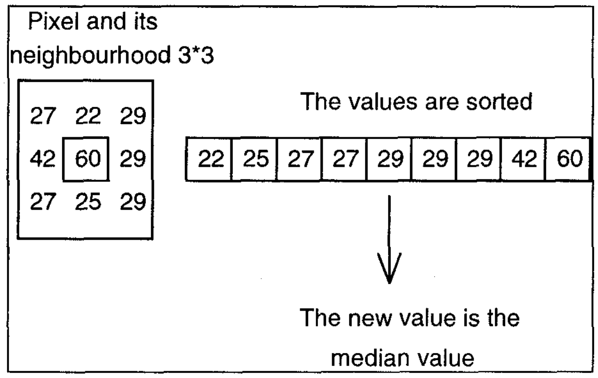

Figure 5. Principle of the median filter

3.2 Extraction of distresses

The purpose of this step is to extract distressed pixels from the image. The algorithm used to detect the distresses is based on the assumptions that: "*a distress is a set of pixels joined together; these pixels are darker than the other ones, and some of them are black or very dark*".

To extract distressed pixels, we use a threshold under which all pixels are considered as distressed. Since the gray level of the background pavement is not the same in different areas of the image, it is necessary to use local thresholds to extract distresses correctly; so, the image is divided into tiles of 40*40 pixels; calling then m the mean value, σ the standard deviation of the gray levels in the tile, and e such as $e = \frac{\sum e_c * v}{\sum e_c}$, e_c being the maximum deviation of each pixel from its neighbors in the tile, and v the value of the central pixel, we use two thresholds defined by $v_1 = \min(m - a_1 * \sigma,\ a_2 * e)$ and $v_2 = \min(m - b_1 * \sigma,\ b_2 * e)$. Coefficients a_1, a_2, b_1, b_2 are settled by experience; v_1 is used to select the dark pixels and v_2 the black or very dark pixels.

The image is then scanned line by line and two sets are created. The first set is filled with the pixels having a level lower than $v1$; the second set is a subset of the first, with the pixels having a level lower than $v2$. Practically, the first set contains the dark pixels and the second the black or very dark pixels. Adjacent pixels are then grouped to build (possible) distressed objects. Only the objects with a number of pixels larger than a minimum are considered as distresses.

At this step, several objects have been identified; the adjacent objects are then clustered to compose larger objects and excluding small objects that are not distresses. The proximity of two objects is determined by the distance between of the nearest pixels of the objects and the size of the objects. It is assumed that for two objects, the bigger and the nearer they are, the more it is consistent to consider that they belong to the same distress. Then, the neighborhood is defined by:

$$neighborhood = d * \frac{a}{size_1 * size_2}$$

where d is the distance between the objects, a is a coefficient, and $size_1$ and $size_2$ are the sizes of the objects.

The algorithm to group the pixels in objects is given in figure 6, below; the set of objects is a list of objects which are themselves lists of pixels.

This method was applied on many images with different distresses. Figures 7 presents several results on images containing transversal cracks, longitudinal cracks and alligator cracks. This tends to show the good quality of our methods.

- Initialize the set of objects to ∅,
- Pass through the image, line by line from the left to the right and the bottom to the top,
  - for each pixel lower than the threshold,
    - Pass through the set of objects,
      - Pass through each object,
        - If the current pixel is adjacent to a pixel of the current object then
          - If this the first object where this happened,
            then add the current pixel to the subset and mark the subset
          - Else add all the pixels of the object to the marked subset,
    - If the pixel has no neighbor object, create a new object and add the current pixel.

Figure 6. The algorithm used to build objects from pixels

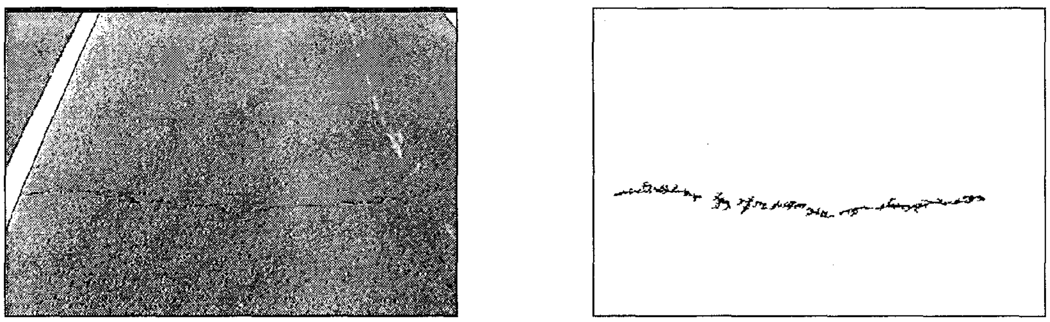

Figure 7a. One transversal crack

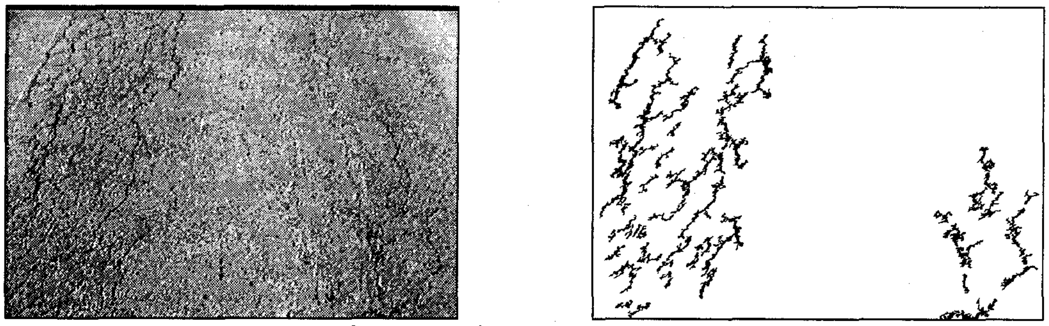

Figure 7b. One alligator crack

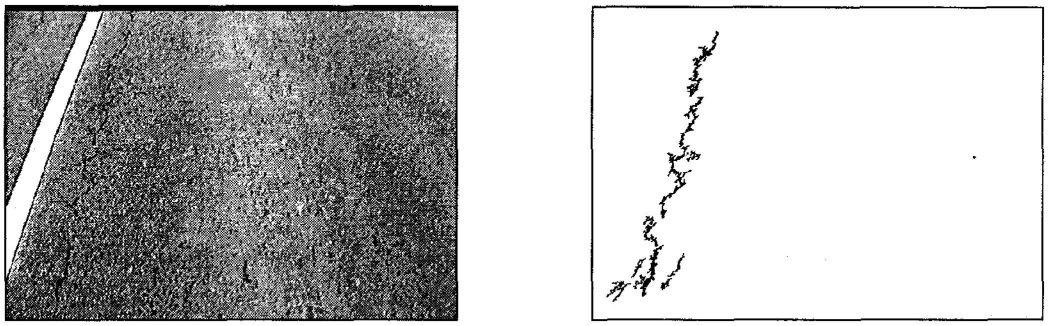

Figure 7c. One longitudinal crack

4. Distresses identification

Confronted with the diversity of neural network architectures [3], [6], [9], [12], we use one back-propagation neural network [11] to identify and classify the different types of distresses. The structure of this network is composed of three layers. The output layer represents the types of distresses that can be identified; four types have been selected: no distress, transversal crack, longitudinal crack, alligator crack. The input layer represents the input parameters. The problem, here, is that after the pre-processing, the number of objects in each image is not the same. To overcome this difficulty, we have decided to consider only the N more important objects in each image. The importance of an object is given by its number of pixels, but we have used several parameters for each object:

- the number of pixels, N_p,
- the position of the center, $X_c = \frac{\sum x}{N}$, $Y_c = \frac{\sum y}{N}$,
- parameters depending of x and y such as: $\sum x, \sum y, \sum xy, \sum x^2, \sum y^2\ \sum x^2 y\ \sum xy^2$,
- the slope, $A = \dfrac{\sum xy - \frac{\sum x \sum y}{N}}{\sum x^2 - \frac{(\sum x)^2}{N}}$,
- inertias: $I_x = \sum x^2 - X_c \sum x$, $I_y = \sum y^2 - Y_c \sum y$ and $I_{yx} = \sum xy - X_c \sum y$
- the correlation, $R = \sqrt{\dfrac{\left(\sum xy - \frac{\sum x \sum y}{N}\right)^2}{\left(\sum x^2 - \frac{(\sum x)^2}{N}\right)\left(\sum y^2 - \frac{(\sum y)^2}{N}\right)}}$,

5. Evaluation of performances

5.1 Rates of correct identification

Our test set is composed of 155 images. These images consist of 48 transversal cracks, 30 longitudinal cracks, 54 alligator cracks and 23 non-distressed images. Among this set, 77 images have been selected to constitute the training set. This set contains 22 transversal cracks, 17 longitudinal cracks, 27 alligator cracks and 11 non distressed images. The table 1 below shows the percentage of good identification obtained on the total data set, when using several parameters and considering a varying number of objects selected as the most important. We can see that all results are good, particularly in the third case.

Nb of objects	Parameters	Trans.	Long.	Allig.	Non-distr.
4	N_p, A, R, X_c, Y_c	88	93	98	91
4	$N_p, \sum x^2, \sum y^2, X_c, Y_c$	94	97	98	96
4	N_p, I_x, I_y, X_c, Y_c	92	100	100	100
4	I_x, I_y, X_c, Y_c	96	87	98	100
5	N_p, A, R, X_c, Y_c	88	83	96	96
5	$N_p, \sum x^2, \sum y^2, X_c, Y_c$	92	97	98	96
5	$N_p, \sum x^2, \sum y^2, X_c, Y_c$	90	100	96	96
5	I_x, I_y, X_c, Y_c	90	93	100	91

Table 1. Rates of good identification

5.2 Processing times

The processing times are rather good in the learning phase: they are no perceivable by the user. However, the identification phase is a little too slow (2 to 3 seconds) in the context of our experiments; it may then be remarked that our experiments are simulations done on a Personal Computer. So, using a faster processor should quicken the identification process, and moreover in the final stage of the realization, the simulation of the neural network could be replaced by one hardware realization using specific components, to improve more these processing times.

6. Conclusion

The method developed here shows that neural networks can be used to identify distresses on a video sequence of pavement images. In comparison with other methods, clustering the pixels as objects seems a good way to improve the identification and to reduce the noise. Our tests give good rates of identification, the learning times are fair, and if the exploitation times are a little too long, they can be easily improved by hardware. In the next stage of our research, a more important set of images will be used to validate the threshold method; others parameters will also be tested to improve the identification step.

References

[1] Acosta J. A., Figueora J. L., and Mullen R. L., "Low-cost Video Image Processing System for Evaluating Pavement Surface Distress", Transportation Research Record 1348, 1992, pp. 63-72.

[2] Bhanu B. and Faugeras O., "Segmentation of Images having Unimodal Distributions", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4, 1982, pp. 408-419.

[3] Bishop, C.M. "Neural Networks for Pattern Recognition", Oxford University Press (1995).

[4] Chou Jaching, O'Neill Wende A. and Cheng Hengda, "Pavement Distress Evaluation Using Fuzzy Logic and Moments Invariants", Transportation Research Record 1505, 1995, pp. 39-46.

[5] Chou Chia-Pei J. and Liau Tz-Kai, "Application of Image Processing to Pavement Distress Survey", Preprint for Transportation Research Board, 75th Annual Meeting, January 1996.

[6] Hertz, J., Krogh, A., and Palmer, R. "Introduction to the Theory of Neural Computation", Addison-Wesley: Redwood City, California (1991).

[7] Jitprasithsiri S. and Lee H., "Development of a Digital Image Processing Algorithm to Compute a Unified Crack Index for Salt Lake City", Preprint for Transportation Research Board, 75th Annual Meeting, January 1996.

[8] Kaseko Mohamed S. and Ritchie Stephen, "A Neural Network-based Methodology for Pavement Crack Detection and Classification", Transportation Research -C, Vol. 1, No 4, 1993, pp. 275-291.

[9] Pao, Y. H., "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley (1989).

[10] Rosenfeld A. and Smith R.C., "Thresholding using Relaxation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 3, 1981 , pp. 598-606.

[11] Rumelhart, D. E., Hinton, G. E. and Williams R. J. "Learning representations by back-propagating errors". Nature, vol. 323 no 9 (1986), pp. 533-536.

[12] Wasserman P. D. ,"Neural Computing: Theory & Practice" Van Nostrand Reinhold: New York (1989).