Classification of Defects in Sewer Pipes Using Neural Networks

By Osama Moselhi,[1] Fellow, ASCE, and Tariq Shehab-Eldeen[2]

Abstract: Deterioration of underground infrastructure facilities such as sewer pipes poses a serious problem to most developed urban centers today. As distribution piping networks age, they deteriorate and may ultimately fail to fulfill their intended functions. To ensure continuity of services and protect the investment made in these networks, municipalities check their conditions regularly. The current practice that is being followed in those checkup programs is usually time consuming, tedious, and expensive. This paper presents an automated system designed for detecting defects in underground sewer pipes and focuses primarily on the application of neural networks in the classification of those defects. A three-layer (i.e., one hidden layer) neural network has been developed and trained using a back-propagation algorithm to classify four categories of defects, namely cracks, joint displacements, reduction of cross-sectional area, and spalling. A total of 1,096 patterns were used in developing the neural network. An example application is described to demonstrate the use and capabilities of the developed system.

INTRODUCTION

Deterioration of underground infrastructure facilities such as sewer pipes poses a serious problem to most developed urban centers today. Much of the underground infrastructure in use in North America today was built during the 1950s and 1960s (McKim 1997). As networks age, they deteriorate and may ultimately fail to fulfill their intended functions. To ensure continuity of services and protect the investment made in these networks, municipalities check their conditions regularly. Usually, CCTV (closed circuit television) cameras that are either mounted on robots or winched between two manholes (Morici 1997) or, alternatively, cameras with powerful zooming capabilities (such as Aqua Zoom, manufactured by Aqua Data Company) are used during these regular checkups. The purpose of these regular checkups is to scan the inner surface of pipes and provide some details about presence, number, and location of defects, if any. Currently, this process ends up with a videotape that has to be visually analyzed by a human inspector in order to identify and classify these defects.

Interviews with several municipal engineers and consultants in Quebec and Ontario, Canada, were conducted to identify the limitations of the currently used practice. These interviews revealed that the cost of sewer inspection using the above-mentioned technique is about CDN $1–1.5 per linear foot. This process is time consuming, tedious, and expensive. It may also lead to diagnostic errors due to lack of concentration by the human inspectors. Clearly if this identification and classification processes can be automated, significant time and money can be saved by eliminating the manual inspection of the videotape. An automated system for this identification process has recently been proposed by the writers (Moselhi and Shehab-Eldeen 1999) and is briefly described here for continuity.

This paper utilizes the configuration of the automated system, proposed earlier (Moselhi and Shehab-Eldeen 1999), and focuses primarily on the application of neural networks in classification of defects in concrete and clay pipes. A three-layer neural network has been developed and trained using the back-propagation paradigm to classify four categories of defects, namely cracks, joint displacements, reduction of cross-sectional area, and spalling. An example application is described to demonstrate the use and capabilities of the developed system.

BACK-PROPAGATION NEURAL NETWORKS

Neural networks were first introduced by McCulloch and Pitts in 1943 (Fausett 1994). These networks attempt to build computer systems that are designed to mimic the human brain (Anderson 1995). Numerous applications of neural networks in civil engineering have been reported in the literature (Moselhi et al. 1991, 1992; and Flood and Kartam 1994a,b). There are different types of neural networks. Each type is considered to be suitable for certain domains of application (Moselhi 1991). Back-propagation neural networks are recognized for their superior performance in pattern recognition and classification tasks (Kaseko et al. 1994; Nekovei et al. 1995). They are also the most commonly used type of networks in civil engineering (Moselhi 1998; Zhao et al. 1998). Fig. 1 depicts a back-propagation neural network. Essentially, it consists of an input layer, an output layer, and one or more hidden layers. Each layer consists of one or more neurons (e.g., X_1, Y_1, and Z_1 in Fig. 1). These different layers are linked to each other by weighted connections. The weights associated with these connections (i.e., W_{i1} and W_{j1} in Fig. 1) are calculated through the training process and generally represent the network's state of knowledge.

Back-propagation networks gain their problem-solving capabilities by learning from cases encountered, in a similar manner to a human gaining work experience. Those cases are called training examples, where, for each case, the input parameters (i.e., geometrical and statistical attributes) form an input pattern (i.e., a certain type of defect) and the desired output parameters (i.e., the network's response) form an associated output pattern (i.e., classification of that defect). In training these types of networks, the different input patterns are presented to the neurons in the input layer ($\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, . . . , $\mathbf{x}_n$) (Fig. 1). Each input is multiplied by its associated weight (w_{in}) and broadcasts the result to the hidden layer. Upon receiving the result, each neuron in the hidden layer sums its weighted inputs and applies its activation function to compute its output. Those computed outputs are then multiplied by their respective weights (w_{jn}) and sent to the output layer. The output layer processes its received inputs in a similar manner to the hidden layer. Upon completion of these calculations, the network compares its calculated outputs to

[1]Prof. and Chair, Dept. of Build., Civ., and Envir. Engrg., Concordia Univ., Montreal, Canada H3G 1M8. E-mail: moselhi@cbs-engr.concordia.ca

[2]Grad. Asst., Dept. of Build., Civ., and Envir. Engrg., Concordia Univ., Montreal, Canada H3G 1M8. E-mail: tshld@cbs-engr.concordia.ca

Note. Associate Editor: Ilker Adiguzel. Discussion open until February 1, 2001. To extend the closing date one month, a written request must be filed with the ASCE Manager of Journals. The manuscript for this paper was submitted for review and possible publication on April 22, 1999. This paper is part of the ***Journal of Infrastructure Systems***, Vol. 6, No. 3, September, 2000. ©ASCE, ISSN 1076-0342/00/0003-0097–0104/$8.00 + $.50 per page. Paper No. 20738.

J. Infrastruct. Syst. 2000.6:97-104.

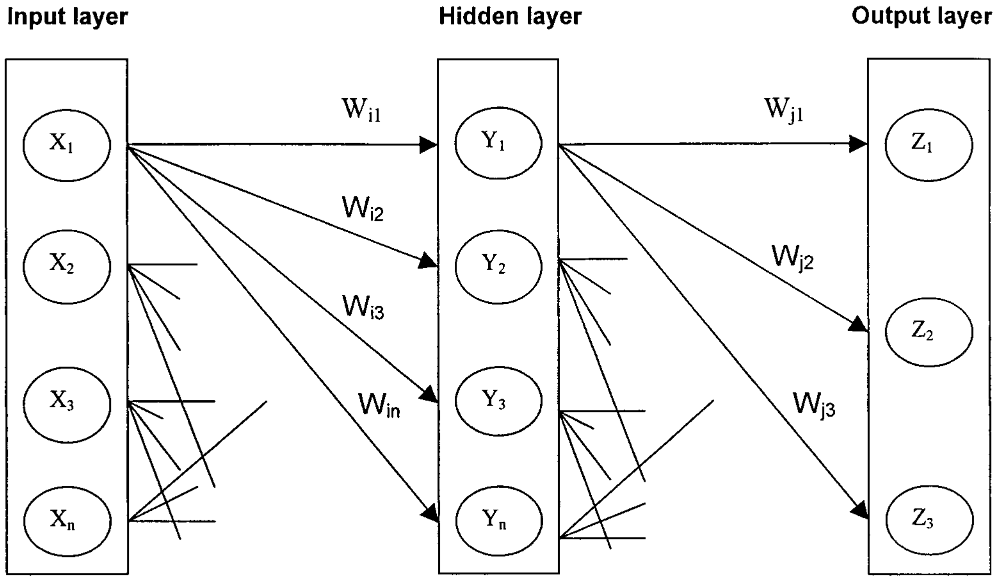


FIG. 1. General Model for Back-Propagation Neural Networks

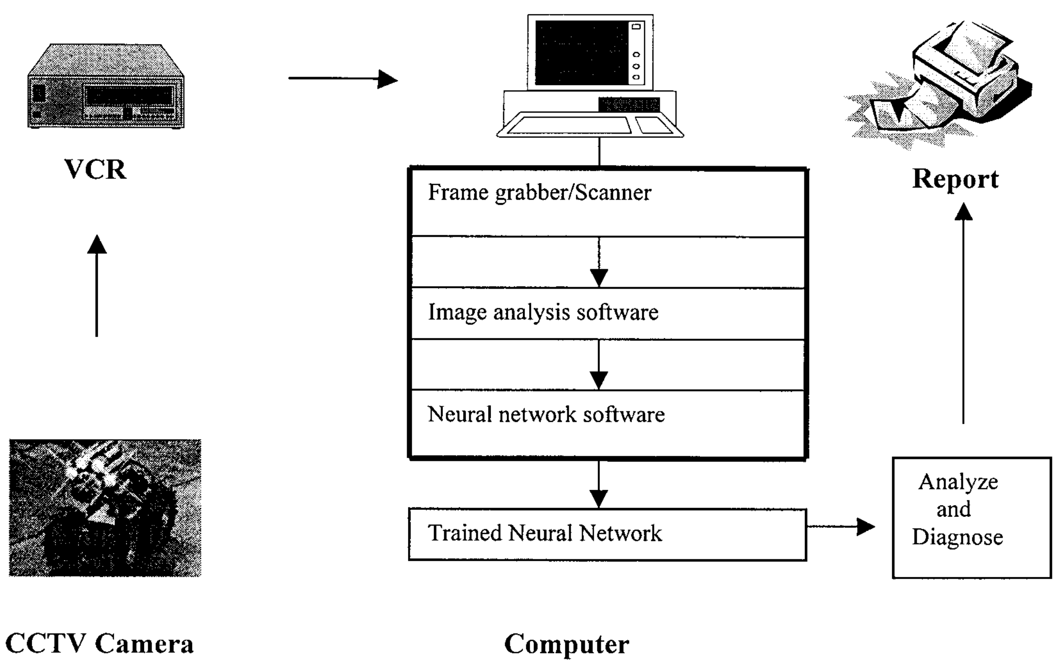


FIG. 2. Schematic Arrangement for Automated System

the desired values, included in the training examples. The weights are then adjusted so as to minimize the difference between the output generated by the network and the desired output. A detailed description of back-propagation networks can be found in a number of textbooks (e.g., Fausett 1994).

PROPOSED MODEL

System Description

Fig. 2 depicts the main components of the proposed system. A CCTV (or Aqua Zoom) camera first scans the inner surface of a pipe and produces a videotape, which is played back using a VCR. The VCR then feeds the information captured on the tape to a computer equipped with frame grabber, image analysis, and neural network software packages (Moselhi and Shehab-Eldeen 1999). The frame grabber captures and digitizes the frames of the acquired images (i.e., a frame per image). The image analysis software analyzes those digitized images and processes them in a manner so as to prepare a suitable input to a neural network. Based on those analyzed images, some feature vectors are extracted, using different image analysis techniques, and are fed to a neural network for training. The trained network can then be used, in lieu of the human inspector, to detect and classify new and similar sets of defects based on their respective extracted features. The processes involved prior to feeding the information to the neural network have been dealt with elsewhere (Moselhi and Shehab-Eldeen 1999) and will not be described herein. This paper focuses primarily on the application of neural networks in the classification of defects in underground sewer pipes. It should be

J. Infrastruct. Syst. 2000.6:97-104.

noted that the system described herein could be adapted for classification of defects in other infrastructure systems, such as pavements. The difference will lie essentially in the set of input attributes and the type of defects.

Data Preparation

The number of input attributes (i.e., number of pixels) needed to represent the video images considered in this model is large (i.e., 512 $\times$ 512 pixels). Reduction of such a huge amount of attributes to a smaller set (i.e., feature vectors) becomes a necessity to help in improving the overall performance of the neural network (Moselhi and Shehab-Eldeen 1999). A feature vector can be defined as a set of geometrical and statistical attributes that describe an object (i.e., defect) and its surroundings in a video image. This reduction has been reported to eliminate, as much as possible, all redundant and irrelevant information that could degrade the classification performance of the network (Looney 1997). It has also been found useful in pattern classification using neural networks (Ritchie 1990; Kaseco et al. 1994; Mashford 1995). The technique basically minimizes the amount of data that has to be fed to a neural network and, accordingly, reduces significantly the number of neurons in the input layer of that network. It ultimately results in improving the learning speed as well as the classification capabilities of the network. In the developed model, instead of dealing with the huge number of attributes in a video image, image analysis techniques (Moselhi and Shehab-Eldeen 1999) were used to generate feature vectors, each consisting of fifteen attributes. These attributes are area, mean density, standard deviation, X-coordinate, Y-coordinate, modal density, perimeter, major axis length, minor axis length, angle, integrated density, modal value of background, minimum gray value, maximum gray value, and the ratio of major axis length to the minor axis length. These attributes were used to represent the various defects to the neural network. A detailed discussion on how these attributes could be measured are found in a number of specialized literature on image analysis techniques (e.g., Moselhi and Shehab-Eldeen 1999). Definitions of these attributes are listed in Appendix I.

Based on the extracted feature vectors, it was noticed that joint displacements might have almost the same attributes as cracks. This is due to the difference in distance between the CCTV camera and both types of defects. In other words, joint displacements away from the camera tend to have similar attributes to cracks closer to the camera. Those similar attributes are small minor axis length, small area, and large ratio of major axis length to minor axis length. The same problem was also faced in differentiating between the further away reduction of cross-sectional area and the nearby small spalling. The only factors that differentiate between the apparently similar attributes, in each of the two situations, are the x and y coordinates (i.e., location), as depicted by the characteristics of the image itself. For example, it was noticed (from the collected sample of video images) that the center of an image is always darker than its surrounding areas (Figs. 3–5). This is due to the fact that the lighting effect vanishes as the distance from the lighting source gets larger. The only objects that are illuminated at this specific area (the center of the image) are the joint displacements and the reduction in the cross-sectional area. This is due to the fact that these two defects tend to project from the surface of the pipe and reflect back the beam of light they are exposed to. Other defects, such as cracks and spalling, do not exhibit the same phenomena. This has been utilized to facilitate the classification process by using a conditional statement that assigns the coordinates of objects located in this dark spot to (1,1) and objects located outside this dark spot to (0,0). This conditional statement was utilized in a spreadsheet application, developed in Microsoft Excel, to prepare the input patterns before feeding them to the neural network. The input file was then completed by assigning the type of defect associated with each pattern. Different techniques for training back-propagation neural networks can be found in a number of textbooks (e.g., Fausett 1994).

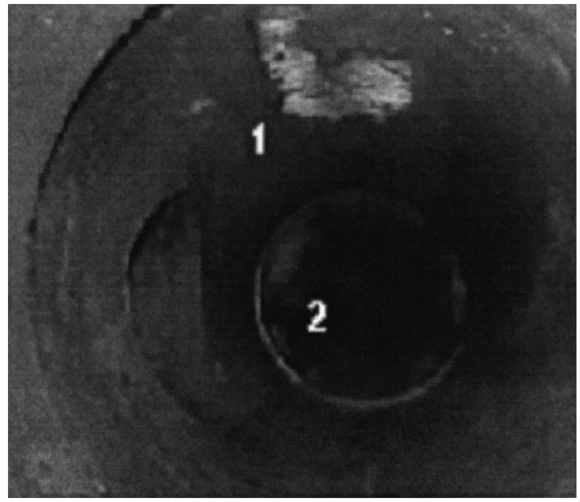


FIG. 3. Sample Pipe

FIG. 4. Sample Pipe

FIG. 5. Sample Pipe

Neural Network Design

In view of the proven capabilities of back-propagation neural networks in classification tasks (Kaseko et al. 1994; Nekovei et al. 1995) and their wide versatility in different civil engineering applications (Moselhi 1998; Zhao et al. 1998), this

J. Infrastruct. Syst. 2000.6:97-104.

approach will be utilized in developing the proposed system. It should be noted that in designing back-propagation neural networks, several questions need to be answered: (1) how many neurons should be in input and output layers; (2) how many hidden layers should be used; (3) how many neurons should be used in each hidden layer; (4) what type of scaling and activation functions should be used to scale the input and output values of the network (NeuroShell-2); and (5) how should input be presented to the network. Probably the most straightforward question to address is the one regarding the number of neurons in input and output layers. Usually the number of neurons in the input layer equals the number of attributes in the feature vector that was selected to represent the input pattern. In the developed model, fifteen neurons were used in the input layer (i.e., one neuron for each attribute).

The number of neurons in the output layer is equal to the desired number of unknown output attributes. The output layer of the developed network consists of four neurons, one for each class of defects. These classes are cracks (defects 3 and 7 in Fig. 4), spalling (defect 1 in Fig. 3), joint displacements (defects 2, 4, 5, 6, and 8 in Figs. 3–5) and reduction of cross-sectional area (defect 9 in Fig. 5). It should be noted that a crack is defined as any object that has a small width as compared with its length and whose width is relatively small. On the other hand, spalling is defined in the context of this paper as any object that has a larger area and minor axis length (width) as compared with cracks. Its length to width ratio is relatively less than those for cracks. Joint displacement is the crescent shape that is developed when two consequent segments of pipes are misaligned. Reduction of cross-sectional area is defined as any object that reduces the cross-sectional area of a pipe at a certain location and, accordingly, obstructs the flow.

To make a decision regarding how many hidden layers should be used, one should consider the overall performance of the network (i.e., its generalization and mapping capabilities) as a final goal. Essentially, the choice is limited to one or two hidden layers. Theoretically, if an infinite number of hidden neurons are used, then three- and four-layer networks (i.e., one or two hidden layers) were reported to have equivalent performance (Tamura and Tateishi 1997). Villiers and Barnard studied the generalization capability of three and four-layer networks for classification tasks. Their conclusion was against the use of four-layer networks in all but the most esoteric applications (Tamura and Tateishi 1997). On the other hand, Obradovic and Yan indicated that four-layer neural networks are superior to the three-layer ones with regard to their mapping capabilities (Tamura and Tateishi 1997). Tamura and Tateishi (1997) found that three-layer networks can exactly represent, with N-1 hidden neurons, any N input-output relationships. In spite of the above differences, it has been documented (Simpson 1996) that three-layer neural networks are sufficient to perform any nonlinear mapping, with only very few exceptions. It was also reported (*NeuroShell-2* 1996) that three-layer networks have been used in 95% of the working applications and that they can be trained much more quickly than four-layer networks. In the developed model, a three-layer network was used (i.e., one hidden layer).

As for the number of neurons that should be used in the hidden layer, it has been documented that selecting the proper number is a matter of trial and error (Hegazy et al. 1994; Loony 1997). In other words, there is no definite number that can be specified beforehand and that guarantees good results. Nevertheless, others (*NeuroShell-2* 1996; Loony 1997) have suggested a number of formulas that give an approximate number of neurons in a hidden layer. The following equation has been applied in selecting the preliminary 32 neurons in the hidden layer (*NeuroShell-2* 1996):

$$N = 0.5(X + Y) + \text{sqrt}(Z) \tag{1}$$

where N = number of neurons in the hidden layer; X = number of input patterns; Y = number of output patterns; and Z = number of patterns in the training set.

Scaling and activation functions are used to bind the input and output to a specific range that neural networks can deal with efficiently. Activation functions are also used to produce outputs at both the hidden and output layers, respectively. Scaling functions are usually applied to the input neurons, while activation functions are applied to the hidden and output neurons. The bounded range in both cases is usually between 0 and 1 or -1 and 1. Those functions are either linear or nonlinear. Some of the most commonly used nonlinear functions are: sine = sin (x); hyperbolic tangent = tanh (x); logistic = $1/[1 + \exp(-x)]$; and gaussian = $\exp(-x^2)$ (*NeuroShell-2* 1996).

Selecting the best activation and/or scaling function for analyzing data is also a matter of trial and error. Some guidelines can only be suggested, such as to scale the input data to -1–1 rather than 0–1 whenever sine or hyperbolic tangent activation functions are used in the hidden layer. The logistic function has also been recommended whenever the outputs provide classification, similar to the case at hand (*NeuroShell-2* 1996). In the developed model, Gaussian (Fig. 6) and logistic (sigmoid logistic) (Fig. 7) activation functions were used in the hidden and output layers, respectively. The inputs were also scaled to (0,1).

Model Training

The developed three-layer back-propagation network was trained to classify four different types of defects, as described in the previous section. The network was developed and trained using the NeuroShell-2 software package (*NeuroShell-2* 1996).

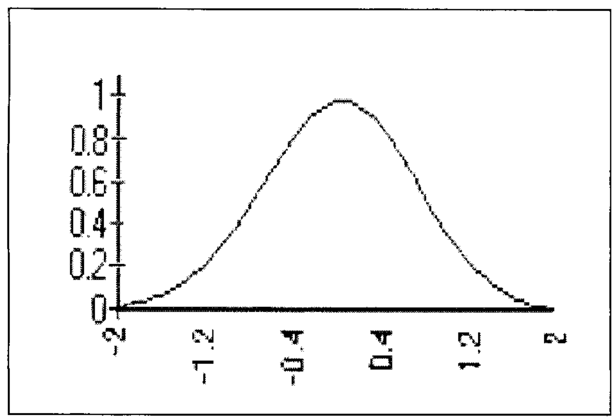


FIG. 6. Gaussian Activation Function

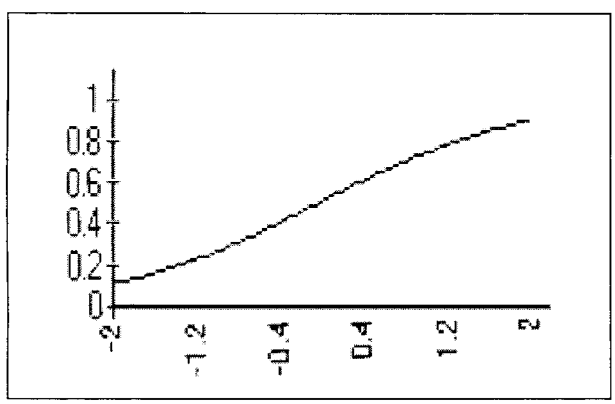


FIG. 7. Logistic Activation Function

J. Infrastruct. Syst. 2000.6:97-104.

The process was implemented on a Pentium II computer with a 233 MHz processor and 64 MB of RAM. The data was provided by a sewer rehabilitation contractor working in Montreal, Canada. A total of 1,096 patterns were used in developing the network. The total number of patterns was randomly divided as follows: 660 patterns (60%) for training; 218 patterns (20%) for testing; and 218 patterns (20%) as a production set. The testing set is a set of patterns that are used to test the generalization capabilities of the network while in training. In so doing, the training process temporarily stops every prespecified and variable number of training iterations (calibration interval) and computes the average error for the training set. The production set is a set of patterns that was not exposed to the network before and is used to test the performance of the trained network. It should be noted that the criteria for stopping the training process was set as shown in Table 1.

It should be noted that, by saving the trained network at the best test set and limiting the calibration interval to 50, overtraining of the network is monitored and prevented. Various combinations of hidden neurons and activation and scaling functions were tried, and the near optimum design was found to be 15 neurons in the input layer, 30 neurons in the hidden layer, and 4 neurons in the output layer. A linear scaling function and Gaussian and logistic activation functions were selected for the input, hidden, and output layers, respectively. The results obtained using the developed network are shown in Table 2. These results are for the 218 patterns not seen by the network during training.

It should be noted that, except for one defect (the reduction of cross-sectional area), good correlation coefficients have been obtained using the trained network (see Table 2). In an effort to improve the performance of the network, a sensitivity analysis was carried out to study the effect of reducing the number of attributes on the overall performance of the network. The general performance of the network was measured by the values of the coefficient of multiple determination (R^2) and the correlation coefficient (r) (*NeuroShell-2* 1996). In this analysis, several networks with different input attributes were developed and their performance compared. Based on the analysis of the results obtained, six attributes were used in the input layer of the developed network. These attributes are area, X-coordinate, Y-coordinate, major axis length, minor axis length, and the ratio of the major axis length to the minor axis length. Fig. 8 depicts the contribution values for the selected attributes. The developed network was tested on the production set (the 218 cases, not seen by the network during training). The developed network was able to correctly classify 214 cases, indicating a success rate of 98.2%. The results, shown in Table 3, depict noticeable improvement in the performance of the developed network, particularly for its ability to detect and classify the ‘‘reduction of cross-sectional area’’ defect. It should be noted that the training time was observed and recorded to be four minutes and 52 seconds.

TABLE 1. Stopping Training Criteria

Criterion (1)	Value (2)
Average error of test set	<3%
Automatically save training as	Best test set
Calibration interval	50

TABLE 2. Results of Preliminary Network

Output (1)	Crack (2)	Spalling (3)	Joint displacement (4)	Reduction of cross-sectional area (5)
R^2	0.8705	0.8804	0.9349	0.4018
Mean squared error	0.030	0.030	0.006	0.008
Mean absolute error	0.067	0.065	0.023	0.021
Minimum absolute error	0	0	0	0
Maximum absolute error	1.00	1.00	0.857	0.861
Correlation coefficient (r)	0.9358	0.9406	0.9678	0.6786

TABLE 3. Results of Final Network

Output (1)	Crack (2)	Spalling (3)	Joint displacement (4)	Reduction of cross-sectional area (5)
R^2	0.9180	0.9238	0.9658	0.8024
Mean squared error	0.019	0.019	0.003	0.003
Mean absolute error	0.042	0.040	0.012	0.011
Minimum absolute error	0	0	0	0
Maximum absolute error	1.00	1.00	0.693	0.619
Correlation coefficient (r)	0.9590	0.9617	0.9831	0.8989

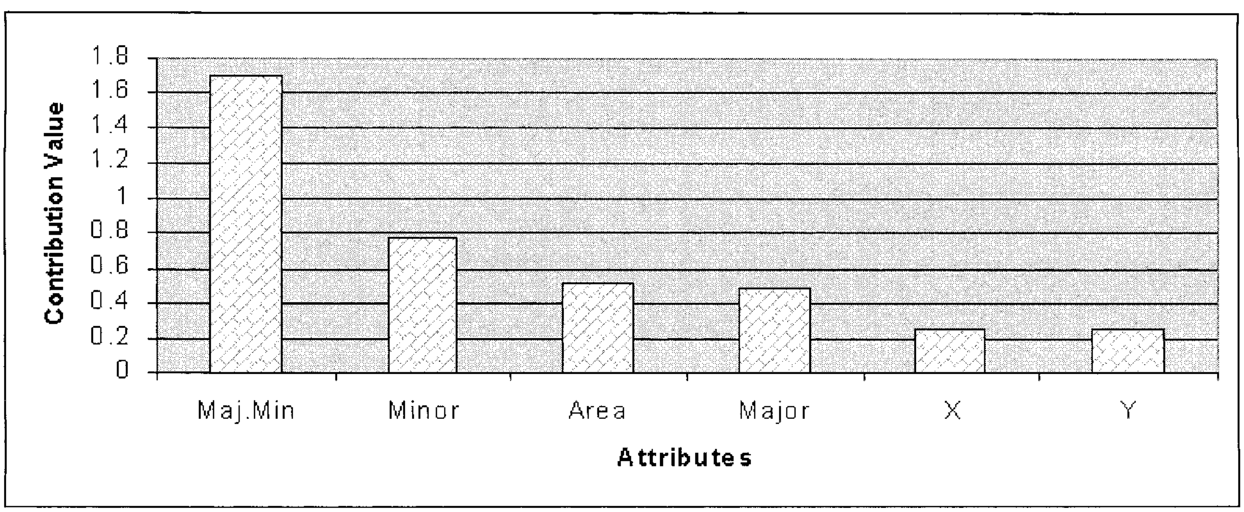
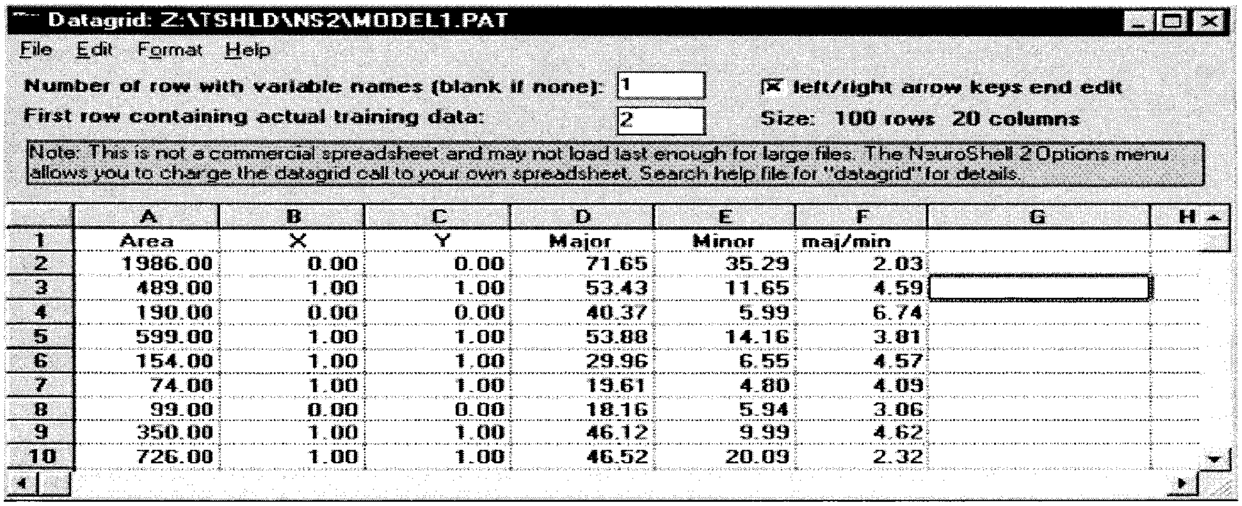


FIG. 8. Contribution Values for Selected Variables

J. Infrastruct. Syst. 2000.6:97-104.

EXAMPLE APPLICATION

In order to demonstrate the use and capabilities of the developed network, three cases (i.e., images), not used in the training process, were selected. The first image (Fig. 3) contains spalling and joint displacement, the second image (Fig. 4) contains two cracks and 3 signs of joint displacement, and the third image (Fig. 5) contains joint displacement and reduction of cross-sectional area. The three images were processed and the input file was prepared (Fig. 9). It should be noted that the rows in this table correspond to actual defects that were detected by the image analysis software. The input



Datagrid: Z:\TSHLD\NS2\MODEL1.PAT

File Edit Format Help

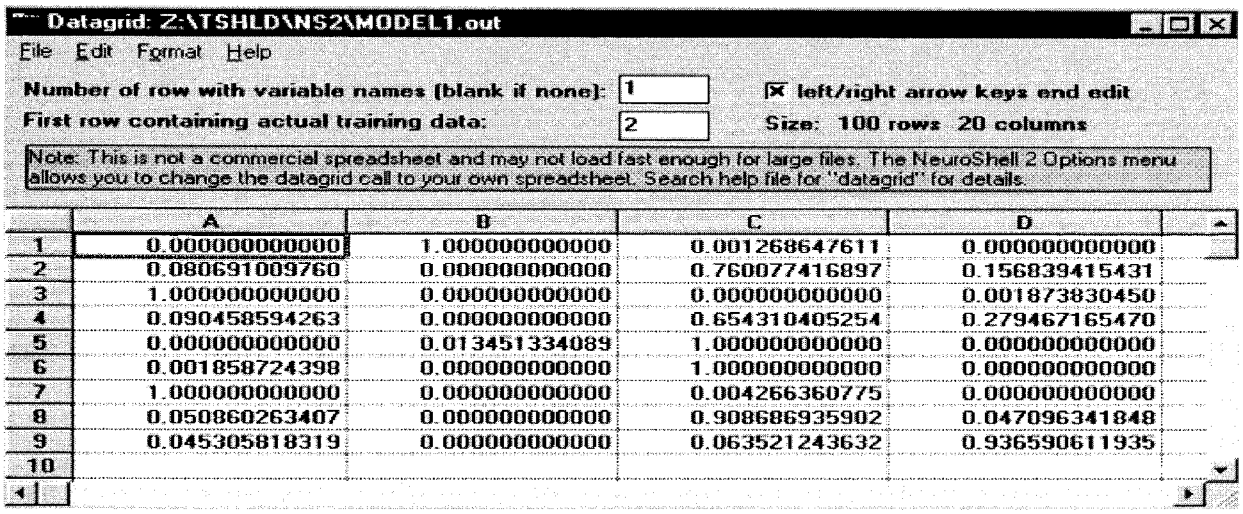
Number of row with variable names (blank if none): 1 | left/right arrow keys end edit

First row containing actual training data: 2 | Size: 100 rows 20 columns

Note: This is not a commercial spreadsheet and may not load fast enough for large files. The NeuroShell 2 Options menu allows you to change the datagrid call to your own spreadsheet. Search help file for "datagrid" for details.

	A	B	C	D	E	F	G	H
1	Area	X	Y	Major	Minor	maj/min		
2	1986.00	0.00	0.00	71.65	35.29	2.03		
3	489.00	1.00	1.00	53.43	11.65	4.59		
4	190.00	0.00	0.00	40.37	5.99	6.74		
5	599.00	1.00	1.00	53.88	14.16	3.81		
6	154.00	1.00	1.00	29.96	6.55	4.57		
7	74.00	1.00	1.00	19.61	4.80	4.09		
8	99.00	0.00	0.00	18.16	5.94	3.06		
9	350.00	1.00	1.00	46.12	9.99	4.62		
10	726.00	1.00	1.00	46.52	20.09	2.32		

FIG. 9. Input File



Datagrid: Z:\TSHLD\NS2\MODEL1.out

File Edit Format Help

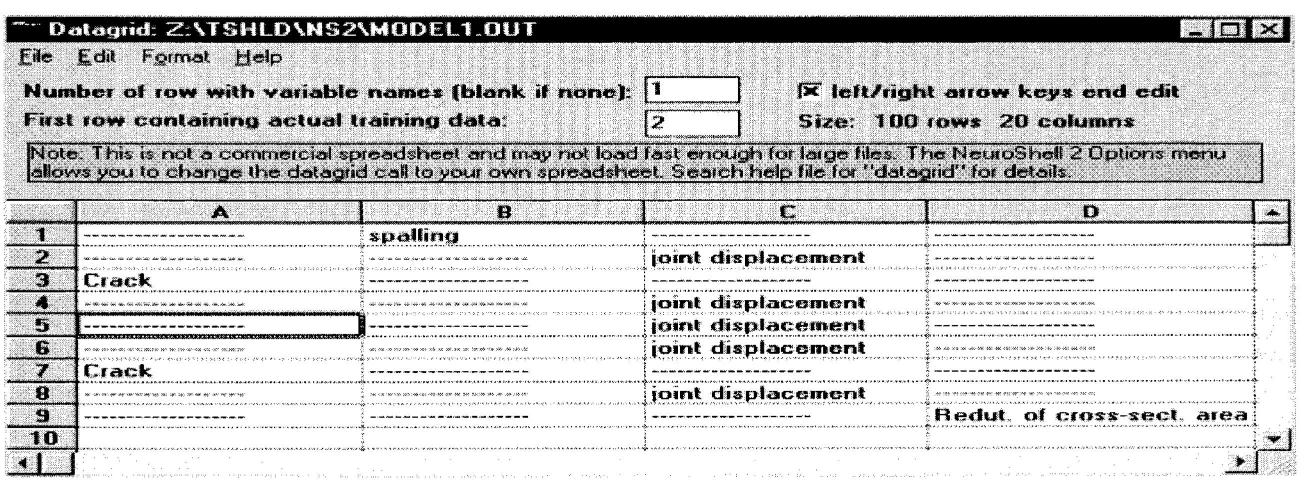
Number of row with variable names (blank if none): 1 | left/right arrow keys end edit

First row containing actual training data: 2 | Size: 100 rows 20 columns

Note: This is not a commercial spreadsheet and may not load fast enough for large files. The NeuroShell 2 Options menu allows you to change the datagrid call to your own spreadsheet. Search help file for "datagrid" for details.

	A	B	C	D
1	0.000000000000	1.000000000000	0.001268647611	0.000000000000
2	0.080691009760	0.000000000000	0.760077416897	0.156839415431
3	1.000000000000	0.000000000000	0.000000000000	0.001873830450
4	0.090458594263	0.000000000000	0.654310405254	0.279467165470
5	0.000000000000	0.013451334089	1.000000000000	0.000000000000
6	0.001858724398	0.000000000000	1.000000000000	0.000000000000
7	1.000000000000	0.000000000000	0.004266360775	0.000000000000
8	0.050860263407	0.000000000000	0.908686935902	0.047096341848
9	0.045305818319	0.000000000000	0.063521243632	0.936590611935
10				

FIG. 10. Output File



Datagrid: Z:\TSHLD\NS2\MODEL1.OUT

File Edit Format Help

Number of row with variable names (blank if none): 1 | left/right arrow keys end edit

First row containing actual training data: 2 | Size: 100 rows 20 columns

Note: This is not a commercial spreadsheet and may not load fast enough for large files. The NeuroShell 2 Options menu allows you to change the datagrid call to your own spreadsheet. Search help file for "datagrid" for details.

	A	B	C	D
1	------------------	spalling	------------------	------------------
2	------------------	------------------	joint displacement	------------------
3	Crack	------------------	------------------	------------------
4	------------------	------------------	joint displacement	------------------
5	------------------	------------------	joint displacement	------------------
6	------------------	------------------	joint displacement	------------------
7	Crack	------------------	------------------	------------------
8	------------------	------------------	joint displacement	------------------
9	------------------	------------------	------------------	Redut. of cross-sect. area
10				

FIG. 11. Processed Output File

J. Infrastruct. Syst. 2000.6:97-104.

file was then processed using the developed network, and the output results are shown in Fig. 10.

As can be noticed from Fig. 10, the output values range from 0 to 1. These values can be considered as the probability that a certain object will belong to any of the four categories of defects. For example, the probability of object number 2 to be classified as a crack, spalling, joint displacement, and reduction of cross-sectional area are 8%, 0%, 76%, and 15.6%, respectively. A threshold value of 50% is considered sufficient for positive classification. As such, if the probability that a certain object belongs to a certain category exceeds 50%, then this object is considered to fall in that category. Although a default value of 50% was used for classification, the developed system allows the user to specify such a threshold value. After defining the selected threshold value to the developed network, the data was processed and the final output results were obtained (Fig. 11). As can be seen from Fig. 11, the developed neural network was able to classify the different defects with 100% accuracy.

CONCLUSION

An automated system for detection and classification of defects in underground sewer pipes has been presented. The system is designed to assist municipal engineers and contractors in diagnosing defects in this class of pipes, with considerable time and cost savings. The developed system utilizes image analysis techniques and neural networks for performing its detection and classification task. The system has the capability to detect and classify four different types of defects: cracks, spalling, joint displacements, and reduction of cross-sectional area.

Image analysis techniques were found to be very useful in preparing data for training neural networks in this domain of application. They provide a useful means for extracting the essential features of these images. As such, they reduce the huge number of attributes associated with video images (such as number of pixels and their gray level values) to a considerably smaller set of features that can be easily handled by neural networks. Six different features were found to have a significant and sufficient contribution toward the classification of defects. These features are area, X-coordinate, Y-coordinate, major axis length, minor axis length, and the ratio of major axis length to minor axis length. A total of 1,096 patterns were used in developing and testing the performance of the developed system. The results obtained from the developed network were found to be good, indicating a high degree of accuracy in classifying the various defects. The developed network was able to correctly classify 214 out of 218 cases, indicating a success rate of 98.2%.

APPENDIX I. DEFINITIONS

Gray value: the brightness value of a pixel (0 for black, 255 for white).

Pixel: picture element (Fig. 12).

Area: area of defect.

Mean density: average gray value of all pixels within the defect.

Standard deviation: standard deviation of the gray values referred to in ‘‘mean density’’ above.

X-Y coordinate: X-Y coordinates of the center of defect.

Modal value: most frequently occurring gray value referred to in ‘‘mean density’’ above.

Perimeter: Parameter of the ‘‘area’’ referred to above (Fig. 12).

Major axis length: length of the major axis of the ‘‘area’’ referred to above (Fig. 12).

Minor axis length: length of the minor axis of the ‘‘area’’ referred to above (Fig. 12).

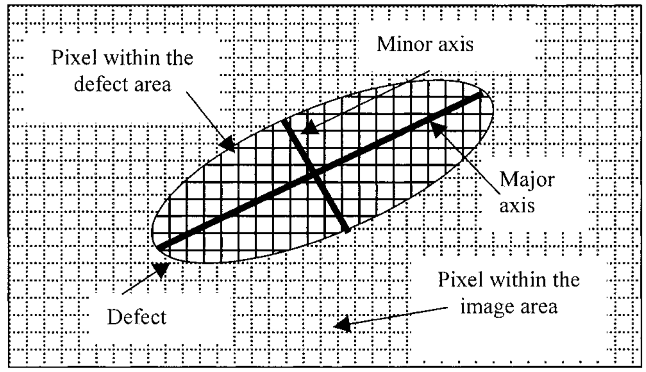


FIG. 12. Geometrical Attributes

Angle: angle between the major axis and a line parallel to the x-axis of the image.

Integrated density = N^* (mean density−modal value of background) where N = number of pixels within the area of the defect.

Modal value of background: most common gray value of image background.

Minimum gray value: minimum gray value within the defect.

Maximum gray value: maximum gray value within the defect.

Ratio of major axis length to minor axis length: major axis length (as defined above)/minor axis length (as defined above).

ACKNOWLEDGMENT

The writers wish to acknowledge the cooperation of Gilles St. Denis, consultant; Leo Gohier, director of the water and wastewater division of the regional municipality of Hamilton, Ontario, Canada; and Antonio Pistilli, project manager of sewer rehabilitation projects for Enviroform, Inc., for providing the data used in this study. The financial support provided by the National Science and Engineering Research Council of Canada is greatly acknowledged.

APPENDIX II. REFERENCES

Anderson, J. (1995). *An introduction to neural networks*, MIT Press, Cambridge, U.K.

Fausett, L. (1994). *Fundamentals of neural networks, architecture, algorithms and applications*, Prentice-Hall, Englewood Cliffs, N.J.

Flood, I., and Kartam, N. (1994a). ‘‘Neural networks in civil engineering. I: Principles and understanding.’’ *J. Comp. in Civ. Engrg.*, ASCE, 8(2), 131–148.

Flood, I., and Kartam, N. (1994b). ‘‘Neural networks in civil engineering. II: Systems and applications.’’ *J. Comp. in Civ. Engrg.*, ASCE 8(2), 149–162.

Hegazy, T., Fazio, P., and Moselhi, O. (1994). ‘‘Developing practical neural network applications using back-propagation.’’ *Microcomp. Civ. Engrg.*, Cambridge, U.K. 9(2), 145–159.

Kaseco, M., and Ritchie, S. (1994). ‘‘Comparison of traditional and neural classification for pavement-crack detection.’’ *J. Transp. Engrg.*, ASCE, 120(4), 552–569.

Looney, C. (1997). *Pattern recognition using neural networks*, Oxford University Press, New York.

Mashford, J. (1995). ‘‘A neural network image classification system for automatic inspection.’’ *Proc., IEEE Int. Neural Networks Conf.*, Institute of Electrical and Electronics Engineers, New York, 2, 713–717.

McKim, R. (1997). ‘‘Selection methods for trenchless technologies.’’ *J. Infrastruct. Sys.*, ASCE, 3(3), 119–125.

Morici, P. (1997). ‘‘Small cameras: diagnosing sewer lateral quickly and easily.’’ *Trenchless Technol.*, 6(10), 40–45.

Moselhi, O. (1998). ‘‘Neural networks and their applications in civil engineering.’’ *Proc., 1st Int. Conf. on New Information Technol. for Decision Making in Civ. Engrg.*, Ecole de Technologie Superieure, Montreal, 1, 17–27.

Moselhi, O., Hegazy, T., and Fazio, P. (1991). ‘‘Neural networks as tools in construction.’’ *J. Constr. Engrg. and Mgmt.*, ASCE, 117(4), 606–625.

Moselhi, O., Hegazy, T., and Fazio, P. (1992). ‘‘Potential applications of neural networks in construction.’’ *Can. J. Civ. Engrg.*, Ottawa, 19(3), 521–529.

J. Infrastruct. Syst. 2000.6:97-104.

Moselhi, O., and Shehab-Eldden, T. (1999). ''Automated detection of defects in underground sewer and water pipes.'' *Automation in Constr.*, 8(5), 581–588.
Nekovei, R., and Sun, Y. (1995). ''Back-propagation networks configuration for blood vessels detection in angiograms.'' *IEEE Trans. on Neural Networks*, 6(1), 64–72.
NeuroShell-2 reference manual. (1996). Ward Systems Group, Inc., Frederick, Md.
Ritchie, S. (1990). ''Digital image concepts and application in pavement management.'' *J. Transp. Engrg.*, ASCE, 116(3), 287–298.
Ritchie, S., Kaseko, M., and Bavarian, B. (1991). ''Development of an intelligent system for automated pavement evaluation.'' *Transp. Res. Rec. 1311*, Transportation Research Board, Washington, D.C., 112–119.
Simpson, P. (1996). *Neural networks theory, technology, and applications*, Institute of Electrical and Electronics Engineers, New York.
Tamura, S., and Tateishi, M. (1997). ''Capabilities of four-layered feedforward neural networks: four layers versus three.'' *IEEE Trans. on Neural Networks*, 8(2), 251–255.
Zhao, J., Ivan, J., and DeWolf, J. (1998). ''Structural damage detection using artificial neural networks.'' *J. Infrastruct. Sys.*, ASCE, 4(3), 93–101.

J. Infrastruct. Syst. 2000.6:97-104.