Piotr Bojarczak, Piotr Lesiak
Technical University of Radom, Faculty of Transport and Electrical Engineering

APPLICATION OF NEURAL NETWORKS INTO AUTOMATIC VISUAL DIAGNOSTIC OF RAILWAY WOODEN SLEEPERS

Abstract: The paper presents the system composing of neural network and image processing procedures being able to classify wooden sleepers on the basis of their image. Image processing procedures extract salient features of sleeper that are further used by neural network in classification process. The system performance was checked on 100 images of good sleeper and 100 images of bad sleeper. System classification rate was equal to 84% for images not taking part in learning process, and 95% for images taking part in learning process.
Keywords: railway wooden sleepers, image processing, classification process

1. INTRODUCTION

Good condition of wooden sleepers is a crucial problem having a large influence on the safety of railway transport. Every year, poor condition of wooden sleepers poses potentially threat to railway traffic. They can cause train derailment, what in turn can generate tremendous human and financial losses. At present, in Polish Railway Lines skilled persons manually check wooden sleeper's condition. They perform visual inspection of wooden sleepers. Such approach is highly inefficient and dependent of the fatigue of controlling person, one person can check daily very few sleepers. Additionally, do not exist any regulations precisely describing the difference between the good and bad sleeper. The inspection is based on the experience and knowledge of controlling person 14. These disadvantages make authors deal with elaboration of automatic system serving to inspection of wooden sleepers. Needless to say that, it is not a simple task. This system should be immune against pebbles occurring on sleepers and variable texture of sleepers. In order to satisfy such tough requirements, it is necessary to use the fusion of image processing procedures and neural networks. Authors intention is not the creation of complete automatic system, but rather focusing on the choice of the best inspection algorithm and checking its usefulness to automatic inspection of wooden sleepers. This system will consist of hardware and software. In our experiment the hardware is confined to the camera. Pictures of both good and bad sleepers have been taken with this camera. The overall number of pictures being taken is equal to 200, 100 corresponding to good sleepers and 100 corresponding to bad sleepers. The software consists of image processing procedures and classification algorithm (neural network). Image processing procedures also called preprocessing serve to extract the salient features of the sleeper best separating bad sleepers from good sleepers. These features constitute the input to neural network realizing classification process. Classification consists in the assignment of sleeper's image on the basis of its corresponding salient features to one of two groups, the first corresponding to good sleeper or the second corresponding to bad sleeper. The paper is composed of three sections. First section presents preprocessing procedures along with the choice of salient features. Second section describes classification process, the choice of neural network type and classification outcomes of the system. Third section - conclusion summarizes obtained results with focusing on some imperfections of the system.

2. PREPROCESSING PROCEDURES

The right choice of the salient features is a crucial task having a large influence on the system performance. The type of selected features also determines the choice of image processing algorithms using to extract them from sleeper's images. Poor state of sleeper can be characterized by cracks of different size occurring on its surface. Cracks usually start developing on the part of the sleeper between its outer edge and the rail edge. In later stage of their development, cracks progress towards middle point of the sleeper. Therefore it seems to be sufficient to analyze only the part of sleeper between its outer edge and the rail edge. Basing on 7, 15 we decided to choose following three salient features: number of cracks, the length of the longest crack and the width of the longest crack. Thus preprocessing task is to recognize cracks, count them and calculate their length and width 2.

At the beginning of preprocessing, color images have been converted to gray scale images. Preprocessing of gray image is much easier than color image. After converting, the image resolution was also changed to 300x200 pixels. Further, the image is cropped to the size containing only the part of the sleeper between its outer edge and the rail edge. Because all pictures were taken from the same distance form sleepers and metal feet mounting rail to the sleeper are fixed size it is sufficient to detect the rail edges and take into account only this part of the image marked by "W" letter in Fig. 1a.

Canny method 4, 10 has been used to detect the rail edge. Though there are many edge detection methods, this method seems to be the most efficient. Its major advantages are low rate of detection errors (low influence of the noise on the detection process), good edge localization (thickness of detected edge is equal to 1 pixel) and the possibility of the influence on the size of detecting edge. This method searches the local maximum of the gradient of the image. Before calculating the gradient, the image is filtered by the Gaussian filter of the mean equal to 0 and standard deviation σ. This method uses low and high threshold to detect weak and strong edge. The weak edge is only detected when it is connected with the strong edge. The size of detecting edge is controlled by the standard deviation σ and these thresholds. After many experiments, we chose the standard deviation equal to 5 and low and high threshold equal to 0,24 and 0,6 respectively. Fig. 1 shows results of image cropping. Fig. 1a presents the original image, Fig. 1b the rail edge detected by Canny algorithm and Fig. 1c the image after cropping to interesting area.

a) b) c)

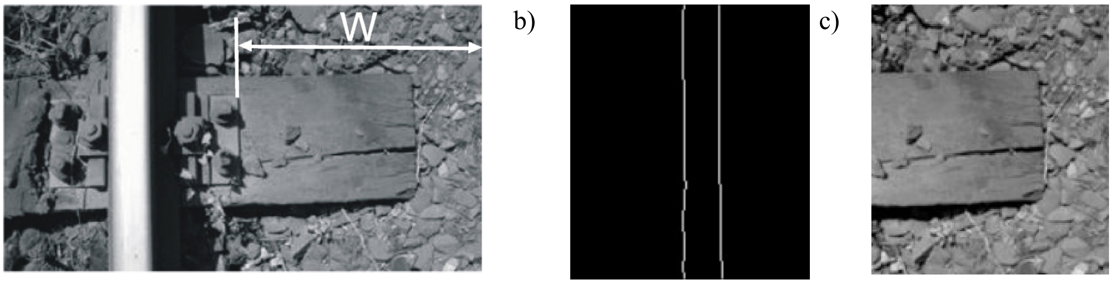

Fig. 1. Results of image cropping: a) the original image of sleeper along with the area undergoing further processing (marked by “W’ letter) b) image with rail edge detected, c) image after cropping to interesting area

Once we have the sleeper’s image confined to interesting area, we have to extract the sleeper from this image. In the Fig. 1c stone chippings surround the sleeper, so that extraction process can be treated as the segmentation of image on the basis of its texture 2, 5. There are many texture recognition algorithms, but we chose the algorithm based on the entropy filter. For the sake of its simplicity and computation efficiency it can be used in situation when analyzing textures differ significantly. This situation occurs in our analyzing images. In order to get rid of small details violating segmentation process, at the beginning each image is blurred with Gaussian filter of the mean value equal to 0 and standard deviation equal to 1. After this, the major entropy filtering process is performed. In this approach the distribution of pixel intensity is calculated in some small area of analyzing image. The size and the shape of this small area are determined by the entropy filter mask. In our experiment we chose the quadratic mask of the side of L = 11 pixels. After determining the distribution of pixel intensity, entropy E corresponding to this distribution is calculated as:

$$E = \sum_{i=0}^{N} P(i) \frac{1}{\log P(i)} \tag{1}$$

where: N is maximal intensity of pixel occurring in the analyzing image (for gray scale image N=255), i is i-th intensity of pixel, $P(i)$ is the probability of occurrence of pixel of i-th intensity in the area of image limited by the entropy filter mask.

During filtration process the mask is successively moved by 1 pixel through the analyzing image and the entropy is calculated for each position of the mask. After the completion of filtration we obtain new image of the same size as initial image but containing the entropy corresponding to small areas of analyzing image. Areas of different texture are represented by different entropy. In last step, the segmentation of image is carried out through determination of entropy threshold T. The area of the image having entropy lower than threshold T is filled with black color and the area having entropy higher than threshold T is filled with white color. In our experiment we assumed threshold T equal to 0.8. Fig. 2 presents segmentation of image both for good and bad sleeper.

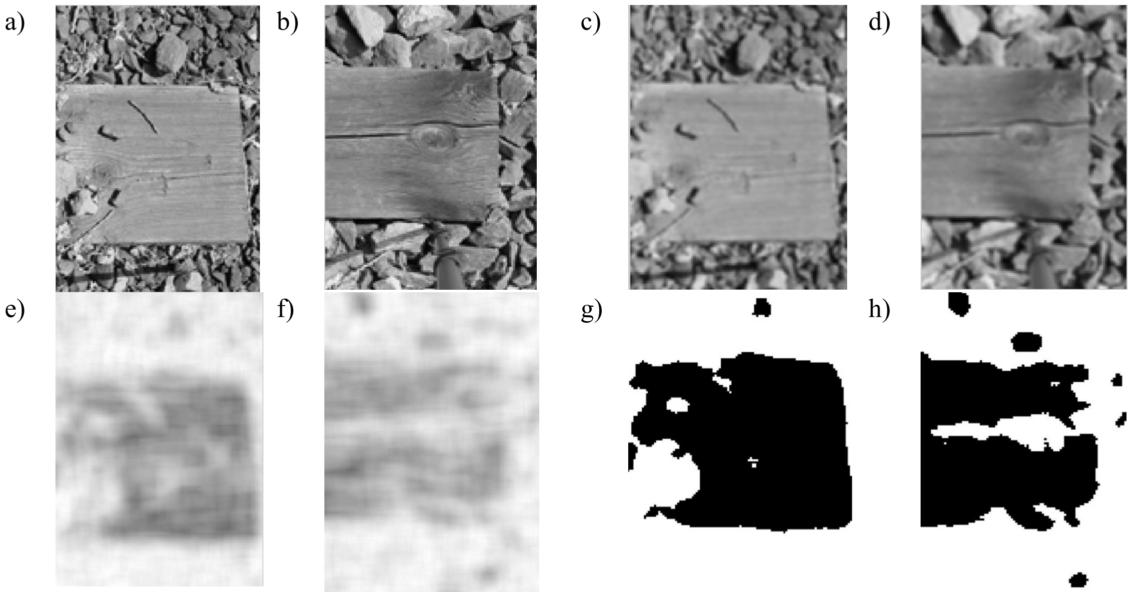

Fig. 2. Segmentation of image: a) image of good sleeper after cropping, b) image of bad sleeper after cropping, c) image of good sleeper after filtering by gaussian filter of mean value equal to 0 and standard deviation equal to 1, d) image of bad sleeper after filtering by gaussian filter of mean value equal to 0 and standard deviation equal to 1, e) image of good sleeper after filtering by entropy filter, f) image of bad sleeper after filtering by entropy filter, g) image of good sleeper after thresholding, h) image of bad sleeper after thresholding

After segmentation, the image changes into binary format in which each pixel is only represented by 0 or 1. Before further processing, 0 (black) is swapped with 1 (white) in the image. After this, the image undergoes sequence of morphological operations serving to remove remained small spurious white objects 3, 5, 6, 10. Fig. 3 shows results of morphological opening with structural element of the size of 7 pixels followed by the selection of objects of the size larger than 400 pixels.

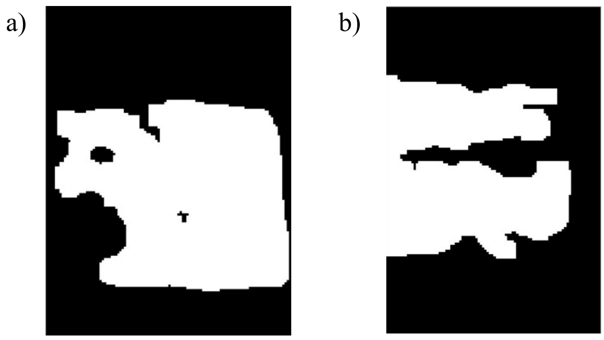

Fig. 3. Images after morphological opening and removing objects larger than 400 pixels:
a) image of good sleeper, b) image of bad sleeper

In order to focus only on the analyzing sleeper, the image is further cropped to the area mostly covered by the analyzing sleeper and the result is enlarged to the size of 300x200 pixels. Fig. 4 presents images of bad and good sleeper after enlargement.

a) b)

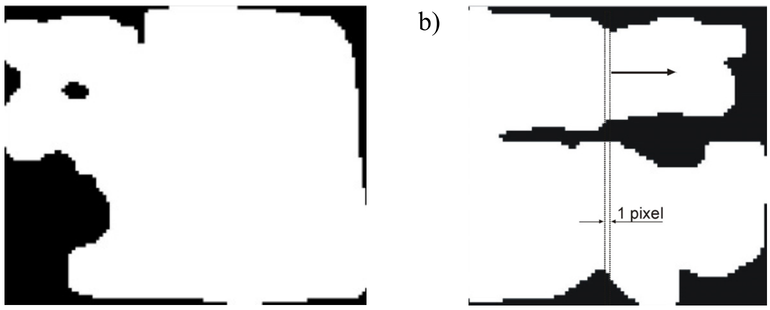

Fig. 4. Images after enlargement: a) image of good sleeper, b) image of bad sleeper along with vertical line using to remove black areas adjacent to horizontal edges of image

Most of arising cracks are placed horizontally or almost horizontally. Basing on this assumption, we can readily remove black areas adjacent to horizontal edges of the image arising from cropping and enlarging it. Fig. 4b illustrates the manner of removing black areas adjacent to horizontal edges.

Vertical line of the width of 1 pixel and the length equal to image height is moved horizontally pixel by pixel through the image. If image pixel occurring in the line is placed in the area adjacent to horizontal edge of image and is black, then it is replaced with white color, otherwise it remains unchanged. Fig. 5 shows images after removing black areas adjacent to horizontal edges of the image.

a) b)

Fig. 5. Images after removing black areas adjacent to horizontal edges of the image a) image of good sleeper, b) image of bad sleeper

The black object existing in Fig. 5b represents the crack. In order to reduce this black object to the skeleton of the width of 1 pixel, skeleton operation is applied to images of Fig. 5. Before this operation, 0 (black color) is swapped with 1 (white color) in the image. Skeleton operation consists in successive removal of pixels on the boundaries of the object without breaking apart this object 5. Fig. 6 presents effects of skeleton operation both for good and bad sleeper.

a) b)

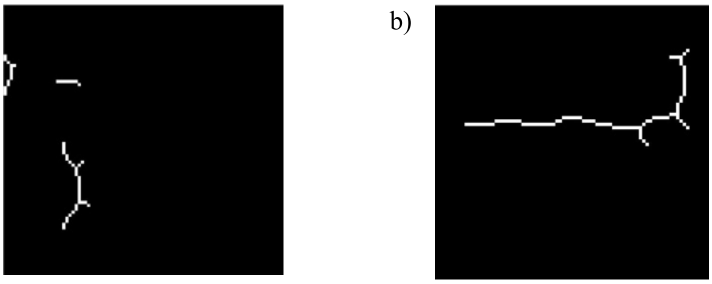

Fig. 6. Images after skeleton operation a) image of good sleeper, b) image of bad sleeper

Apparent in Fig. 6b skeleton represents the crack. This skeleton contains also spurious small “twigs”, that can violate further classification process. These small twigs have been removed with the help of pruning algorithm proposed in 12, 13. This algorithm identifies junction points on the skeleton. Junction point is the point of intersection of lines. At the beginning, junction points being placed near the both ends of the skeleton are removed. After removing these points the skeleton is decomposed into reduced version of skeleton and isolated lines. Isolated line is the line in that neither point is connected with any point of reduced version of skeleton. If isolated line is less than 35 pixels then it is removed. Otherwise, it is again coupled to reduced version of skeleton. Next, new junction points are determined and the process is repeated until there is no change in the skeleton. Fig. 7 shows images after application of pruning.

a) b)

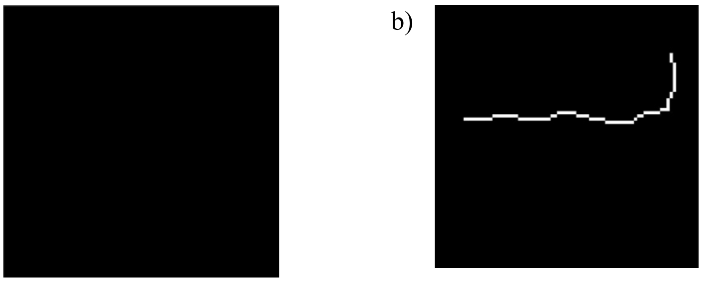

Fig. 7. Images after application of pruning a) image of good sleeper, b) image of bad sleeper

Images of Fig. 7 constitute the base to calculation the number of cracks, the length of the largest crack and the width of the largest crack. The length of the largest crack is the length of the straight line connecting end points of the largest crack. The width of the largest crack is calculated as the ratio of its size (in pixels) to its length expressed as above. Presented here preprocessing stage has been realized in Matlab as the software, which reads sequentially sleeper’s images and generates file with extracted features using in classification process.

3. CLASSIFICATION PROCESS

After extracting salient features, it is necessary to perform classification process. Classification consists in the assignment of sleeper’s image on the basis of its corresponding salient features to one of two groups, the first corresponding to good sleeper or the second corresponding to bad sleeper. Neural networks are commonly used to classification tasks 1, 11. Their ability to make decision on the basis of incomplete data and their self-learning possibility make them popular tools in solving different classification problems. The oldest neural network is Multilayer Perceptron (MLP) 8. It is now a bit obsolete and its learning process is very long. After many experiments, we decided to use hybrid network. Its learning process is much faster than MLP learning process and it requires much less training data for good classification 11. Fig. 8 presents the structure of hybrid network. It consists of Kohonen network followed by MLP network.

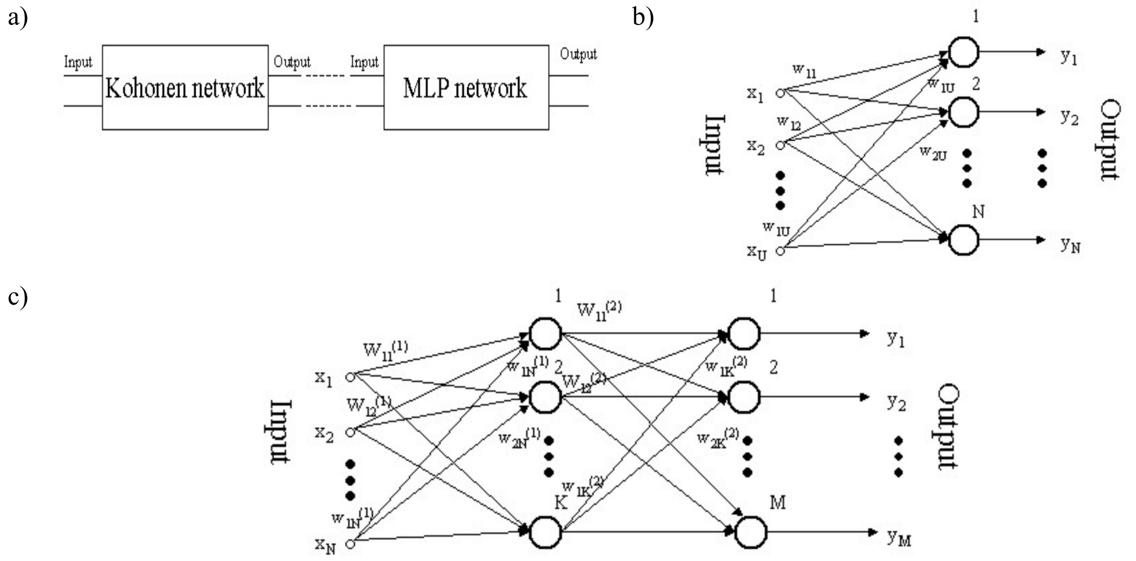

Fig. 8. Hybrid network: a) structure of hybrid network,
b) structure of Kohonen network c) structure of MLP network

Hybrid network is a fusion of Self-Organizing network (Kohonen) and the network being taught in supervised manner (MLP). They consist of neurons (marked as circles). Each output neuron of Kohonen network is connected with each input neuron of MLP network. Teaching process of hybrid network is performed separately for each type of network. At the beginning Kohonen part of the network is taught in an unsupervised manner. During this process input vector **x** corresponding to salient features is given to network's input. Next the winner neuron is determined whose weight vector is in the nearest distance from input vector **x**. The distance between these vectors is Euclidean distance. After determining the winner neuron, its weights and weights of its neighbor neurons are adapted in following manner 9:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta_i G(i, \mathbf{x}_k)[\mathbf{x}_k - \mathbf{w}_i(k)] \tag{2}$$

where: k is adaptation step, $G(i, \mathbf{x}_k)$ is the vicinity function for i-th neuron expressing the degree of following i-th neighbor neuron its winner neuron and η_i is a learning coefficient changing during learning phase from its maximal value (less than 1) at the beginning of learning process to zero.

After finishing the Kohonen learning process, weights of Kohonen part of network reflect distribution of input vector **x**. In second stage of learning process, MLP part of the network is taught in a supervised manner. Output vector of Kohonen part is now the input vector **x** for MLP network. Let $\mathbf{y} \in R^M$ be the output vector generated by MLP, $\mathbf{x} \in R^N$ the input vector of MLP and $\mathbf{d} \in R^M$ desired output vector of MLP. MLP maps the input space R^N to the output space R^M. During learning process each weight vector $\mathbf{w}_i$ of MLP is adapted in the following manner 8:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) + \eta \mathbf{p}_k \tag{3}$$

where: k is adaptation step, η is a learning coefficient less than 1 and $\mathbf{p}_k$ is a directional vector dependent of current value of weight vector $\mathbf{w}_i$. Directional vector $\mathbf{p}_k$ is determined through minimization of the following goal function:

$$E = \frac{1}{2}\sum_{p=1}^{S}\sum_{r=1}^{M}\left[f\left(\sum_{i=0}^{K} \mathbf{w}_{ri}^{(2)} f\left(\sum_{j=0}^{N} \mathbf{w}_{ij}^{(1)} x_{pj} \right)\right) - d_{pr} \right]^2 \tag{4}$$

where: f is neuron activation function, S is the number of learning pair (**x**, **d**) of MLP network, M is the number of output neurons equal to dimension of the output space R^M, N is the number of input neurons equal to dimension of the input space R^N, K is the number of hidden neurons – Fig. 8c.

In our experiment dimension of input vector **x** of Kohonen part is equal to 3 (number of salient features), the number of output neurons in MLP part is equal to 2 (network output vector **y** = [1, 0] corresponds to the image of good sleeper and **y** = [0, 1] to the image of bad sleeper).

We also chose 40 neurons for Kohonen part of hybrid network and 7 hidden neurons for MLP part of hybrid network. The hybrid network has been trained on 70 randomly chosen images of good sleeper and 70 randomly chosen images of bad sleeper (100 images of good sleeper and 100 images of bad sleeper are available). Fig. 9 shows spatial distribution of salient features for images taking part in training process ("+" corresponding to good sleeper, "o" corresponding to bad sleeper) along with distribution of weights of neurons for Kohonen part marked by filled circle.

Thanks to self-organizing feature of Kohonen part, the distribution of salient features has been mapped by weights of neurons of Kohonen part into clusters. Some weights of Kohonen part cover distribution of salient features of images of good sleeper whereas other part of weights covers distribution of salient features of images of bad sleeper. MLP part "staples" all weights (neurons having these weights) covering distribution of salient features of images of good sleeper in first group and all weights (neurons having these weights) covering distribution of salient features of images of bad sleeper in second group.

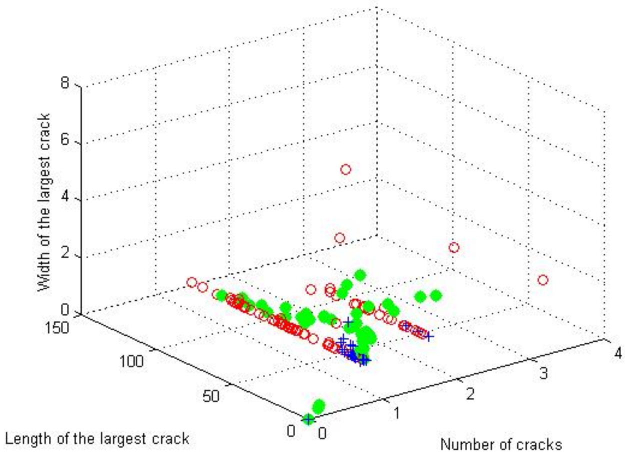

Fig. 9. Distribution of salient features for images taking part in training process along with distribution of weights of Kohonen part

When the input vector to hybrid network is being given after finishing its learning process, this neuron is activated whose weights are nearest to the input vector. If activated neuron belongs to the group of neurons covering distribution of features of images of good sleeper then MLP activates this group and its number is put on the output of hybrid network as the output vector [1, 0]. If activated neuron belongs to the group of neurons covering distribution of features of images of bad sleeper then MLP activates this group and its number is put on the output of hybrid network as the output vector [0, 1].

The performance of the hybrid network was checked on 30 images of good sleeper and 30 images of bad sleeper not taking part in learning process. The classification rate was equal to 84%. It means that 9 sleeper images out of 60 were misclassified. In case of images taking part in learning process this rate raised to 95%.

4. CONCLUSION

Presented here system was able to classify wooden sleepers on the basis of their images. Despite its high classification rate (84%) it suffers from some imperfections. The system has the problem with classification of wooden sleepers when bushes or grass heavily covers them. Also large clusters of pebbles occurring on the sleeper introduce errors in classification. Bushes and grass should not pose a large problem, it is up to railway line staff to maintain the railway line in proper condition. However pebbles are the problem. It seems that application of some algorithm taking into account the real shape of crack in the preprocessing procedures could alleviate this problem. We would like to deal with this problem in our further research.

References

1. Bishop C.M.: Neural networks for pattern recognition, Clarendon press- Oxford, 1997.
2. Bojarczak P., Lesiak P.: Preprocesing w diagnostyce wizyjnej podkładów kolejowych. TRANSCOMP 2009. Logistyka 6/2009 (płyta CD).

3. Bovik A.: Handbook of Image and Video Processing, Academic Press, San Diego USA, 2000.
4. Canny J.: A computation approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, (8) (679-698), 1986.
5. Choraś R.S.: Komputerowa wizja. Metody interpretacji i identyfikacji obiektów. EXIT, Warszawa 2005.
6. Gonzalez A., Woods J.: Digital image processing. Adddison-Wesley, 2003.
7. Gutta G.: Maschine Vision for the automatic classification of image acquired from non destructive tests. Dalarna University College, Departament of Computer Engineering, Sweden 2007.
8. Haykin S.: Neural networks a compresive foundation. Pretince Hall, 1999.
9. Kohonen T.: Self-Organizing Maps. Springer, 1997.
10. Malina W., Smiatacz M.: Cyfrowe przetwarzanie obrazów. EXIT, Warszawa 2008.
11. Mitchell T.M.: Machine learning. McGraw-Hill, 1997.
12. Niemisto A., Dunmire V., Yli-Harja O., Wei Zhang, Shmulevich I.: Robust Quantification of in Vitro Angiogenesis Through Image Analysis. IEEE Transactions on Medical Imaging, (4) (549-553) , 2005.
13. Surhone L.M., Timpledon M.T., Susan F. Marseken S.F.: Pruning (algorithm). Betascript Publishers, January 2010.
14. Warunki techniczne utrzymania nawierzchni na liniach kolejowych Id-1 (D-1). PKP PLK S.A., Warszawa 2005.
15. Yella S., Dougherty M., Gupta N.K.: Condition monitoring of wooden railway sleepers. Transportation Research Part C Elsevier (17) (38-55), 2009.

ZASTOSOWANIE SIECI NEURONOWYCH W WIZYJNEJ AUTOMATYCZNEJ IDENTYFIKACJI USZKODZEŃ PODKŁADÓW KOLEJOWYCH

Streszczenie: W artykule przedstawiono system składający się z sieci neuronowej oraz procedur obróbki obrazów pozwalający na diagnostykę drewnianych podkładów kolejowych na podstawie ich obrazów. Procedury obróbki obrazów wydobywają najistotniejsze cechy podkładów na podstawie których przeprowadzana jest ich klasyfikacja przez sieć neuronową. Poprawność działania systemu została przetestowana na 100 obrazach nieuszkodzonych podkładów oraz 100 obrazach podkładów uszkodzonych. System prawidłowo sklasyfikował 84% obrazów niebiorących udziału w procesie uczenia oraz 95% obrazów biorących udział w procesie uczenia.

Słowa kluczowe: drewniane podkłady kolejowe, przetwarzanie obrazów, klasyfikatory