Neural Network Structure with Alternating Input Training Sets for Recognition of Marble Surfaces

Irina Topalova
Faculty of German Engineering Education and Industrial Management
Technical University of Sofia, Bulgaria
Sofia, Bulgaria
itopalova@abv.bg

Magdalina Uzunova
Department Mathematics
University of Architecture, Civil Engineering and Geodesy
UACEG
Sofia, Bulgaria
magi.uzunova@abv.bg

***Abstract*— The automated recognition of marble slab surface textures is an important task in the contemporary marble tiles production. The simplicity of the applied methods corresponds with fast processing, which is important for real-time applications. In this research a supervised learning of a multi-layered neural network is proposed and tested. Aiming high recognition accuracy, combined with simple preprocessing, the neural network is trained with different alternating input training sets including combination of high correlated and de-correlated input data. The obtained good results in the recognition stage are represented and discussed, further research is proposed.**

Keywords-neural network; recognition; texture; preprocessing.

I. INTRODUCTION

The automated recognition of marble slab surfaces is a fudge factor for increasing the production efficiency. The prerequisite for that is to apply reduced hardware equipment and simple software methods to obtain fast processing in real-time work. Taking into account these requirements, the achieved recognition accuracy is very important especially in the case of similar marble surface textures. Finding the appropriate input data transformations would facilitate the next recognition step. Thus, the choice of simple texture parametrical descriptions and their interclass de-correlation in the preprocessing stage is an essential question. The next one is the right choice of an appropriate trained adaptive recognition structure.

In this research a simple hardware structure combined with a supervised learning of a multi-layered neural network (NN) is proposed and tested. Two different types of texture descriptions are used for training the network. Aiming high recognition accuracy, combined with simple preprocessing, the NN is trained with these alternating input training sets including combination of high correlated and de-correlated input data.

The obtained results, when training the network with a single type and with different types of alternating input training sets are represented. The obtained good results in the recognition stage even for similar textures are represented and discussed. Further research is proposed.

In Section II, the state of the art is represented, together with a discussion about disadvantages of the listed methods concerning the obtained results. In Section III, the selected preprocessing method is explained and the used system components are described. Section IV contains the experimental conditions and results, along with comparative discussions. In Section V, the conclusions and future work are defined.

II. RELATED WORKS

There are many related research proposals for recognition of similar, different shaded or hardly distinguishable marble textures. One of the often investigated proposals for extraction of texture feature descriptions is the statistical, instead of structural methods. In [1], the authors represent texture-based image classification using the gray-level co-occurrence matrices (GLCM) and self-organizing map (SOM) methods. They obtain 97.8 % accuracy and show the superiority of GLCM+SOM over the single and fused Support-Vector-Machine (SVM), over the Bayes classifiers using Bayes distance and Mahalanobis distance. To identify the textile texture defects, the authors in [2], propose also a method based on a GLCM feature extractor. The numerical simulation shows error recognition of 91%. The authors in [3], investigate marble slabs with small gradient of colors and hardly-distinguishable veins in the surface. They apply a faster version of a Co-occurrence matrix to form a feature vector of *mean, energy, entropy, contrast* and *homogeneity*, for each of the three color channels. Thus they constitute a NN input feature vector of 15 neurons and the designed network presents 15 neurons in the input layer. In this case the authors claim high-speed processing and recognition accuracy of 80-92.7%. Another known approach for texture segmentation and classification using NN as recognition structure, is the implementation of Wavelet transform over the image and feeding the network with a feature vector of Wavelet coefficients [4][5]. Training a hierarchical NN structure with texture histograms and their second derivative is also announced as giving good recognition accuracy [6]. Considering the explicated data, we could formulate some disadvantages of the approaches given above. The use of GLCM needs high computations and even faster version of a Co-occurrence matrix as given in [3], needs computations

 ISBN: 978-1-61208-555-5

multiple times over the whole image for each of the three colors. The calculation of Wavelets is also a time-consuming operation. Using hierarchical NN structure, feeding different NNs [6], with different input feature vectors, would be more complicated, particularly for real-time applications in different hardware platforms. The obtained accuracy is not approaching 100%. Thus, the important source of optimizing the recognition method lies in the simplifying the preprocessing stage/the input feature vector and in finding a more efficient training method along with reducing the NN nodes.

III. Method and System Development

In this section, a motivation for choosing the proposed input training sets is given, along with description of the system components.

A. *Selecting a Preprocessing Method*

Complying with the finding that NN training would be more efficient, when applying different types of intra class input data [7][8], we choose to train a single MLP Back-propagation NN alternating with two types of input vectors. The first one is the calculated first derivative dH(g)/dg of the corresponding normalized grey level (g) texture histogram H(g). As we test marble tiles with similar textures, the obtained inter class vectors are high correlated, which will "embarrass" the NN class-separation capabilities. However, we use this training set because it reflects the vertical H(g) axis changes. To compensate the high inter class correlation, we investigated different types of simple mathematical transformations over the H(g), to find de-correlated input training vectors. In our case, U = Exp(k.H(g)) gave the best reduction of the inter class correlation coefficient and was chosen for second input training set. So, the NN is trained with these alternating input training sets including combination of high correlated and de-correlated input data.

B. *System Components*

The proposed test system includes one smart camera *NI 1742(300dpi)* with triggered infrared lighting, software *Vision Builder for Automated inspection AI'14 (VB for AI)* and *Neuro-System V5.0* - shown in Figure1. The images are taken at the same distance with the same spatial resolution. In off-line mode, the captured image contrast quality is improved in *VB for AI* applying simple lookup table, the corresponding preprocessing of the two types of training sets are calculated. In on-line/test mode the same operations are performed for each test sample, but the input data only "go" through the saved (after the training), weight matrix **W**. The results are given to *VB for AI* for visualization and preparation for extraction through standard interfaces.

IV. Experiments and Results

In this section, the details of the preprocessing stage are given, along with a description of the MLP NN training. Also, the choice of the NN parameters is explained. In the end of the section, the achieved results are shown and a comparative analysis is represented.

A. *Preprocessing Stage*

The experiments are carried out for three marble tiles/classes with similar textures given in Figure 2. The color images are transformed to grey level images applying the method (R+B+G)/3, which will reduce and average the color channel information. It is a loss of information, but it will simplify the further calculations. Calculating different color histograms or any color model parameters (as Hue color parameters), aiming to prepare different input vectors for MLP NN, would require a much more complex NN

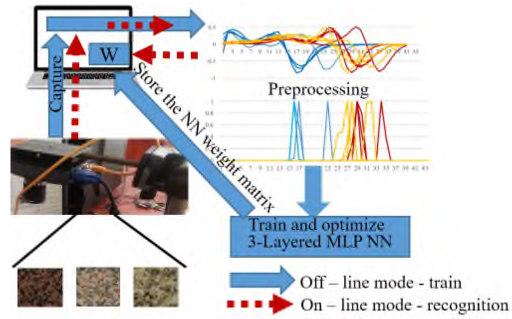


Figure 1. System components

structure. In our case, this loss of information is compensated by using de-correlated input data as Exp(k.H(g)). To evaluate the similarity between samples of

a/ b/ c/

Figure 2. Grey level marble tiles – a/-class1, b/-class2, c/-class3

classes i, j for different input NN feature vector descriptions, the correlation coefficient r_{ij} is calculated according to [9]. Points 1 to 4 of X axis in Figure 3, show the correlation between some exemplars of classes 1 and 2, points 5 to 8 - the correlation between exemplars of classes 2 and 3, points 9 to 12 - the correlation between exemplars of classes 1 and 3. As the coefficient r_{ij} for H(g) varies in the range (-0.24;0.96), it shows very high similarity particularly between classes 2 and 3. That is the reason for searching additional transformations over H(g), to achieve low inter class correlation and better separation between the classes. Thus, the input training vectors will facilitate the NN generalizing capabilities. As the normalized $H(g)/H_{max}(g)$ variables are in the range (0;1), the function U= Exp(k.H(g)), where $k \in \mathbb{R}$, will be suitable, because the correlation coefficient is not invariant about this transformation. Good separable descriptions are obtained when choosing proper values for k (k=10, k=20, k=-10, etc.). With k=100, i.e., for U=Exp(100.H(g)), we achieve the best de-correlation results, shown in Figure 3., where r_{ij} varies in the range (-0.036;0.24). For the normalized H(g) values given in Figure 4, the calculated U are represented in Figure 5. As the function U has a smoothing effect over H(g), it also

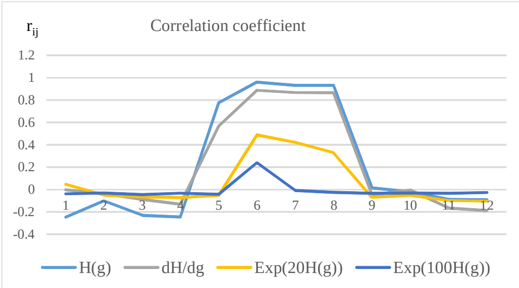


Figure 3. Correlation coefficient r_{ij} for different input training sets

reduces the sharpness of vertical changes in H(g). To conserve and even increase these informative areas we use dH(g)/dg as additional NN training set. It also gives better r_{ij} then H(g). The training set of dH(g)/dg is shown in Figure 6.

B. Train Method

The decision plane consists of a 3-layered MLP NN, trained with well-known Back-propagation algorithm [8]. The input layer has 45 sampled dH(g)/dg and U=Exp(100.H(g)) values over the histograms, calculated in

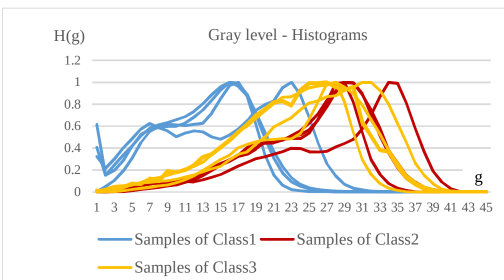


Figure 4. Normalized histogram values H(g) for samples of classes1, 2, 3

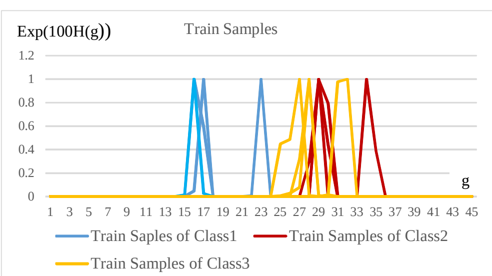


Figure 5. Train Exp(100.H(g)) values for the samples of classes1, 2, 3

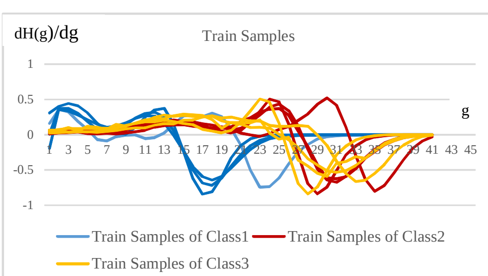


Figure 6. Train dH(g)/dg values for the samples of classes1,2 and 3

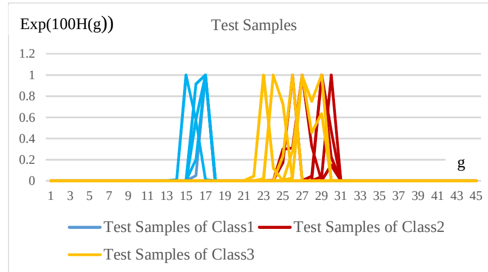


Figure 7. Test Exp(100.H(g)) values for the samples of classes1,2 and 3

VB for AI [10]. They are given alternative to the NN input layer nodes. By training of MLP NN we want to obtain "softer" transitions or larger regions, where the output stays near to "1" or "-1" (using tangent hyperbolic as activation function). The training in off-line was repeated to find the optimized MLP NN structure according to the method given in [5]. We obtained the best fitting structure with 18 hidden

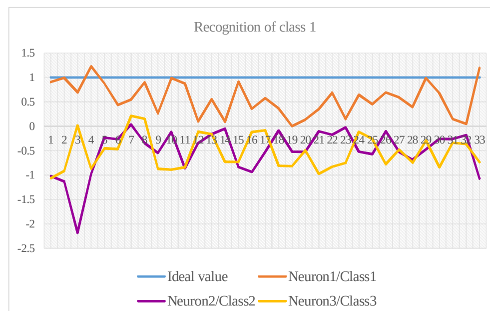


Figure 8. Output neuron values for recognition of class1 samples

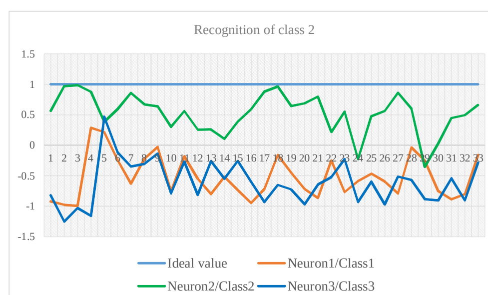


Figure 9. Output neuron values for recognition of class2 samples

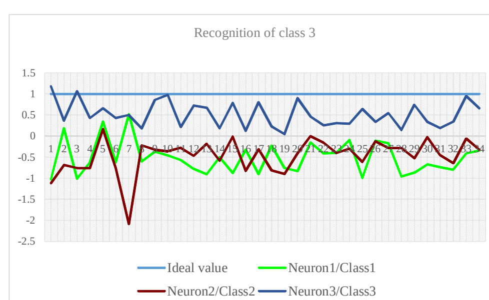
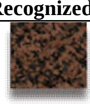
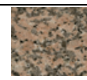
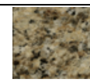


Figure 10. Output neuron values for recognition of class3 samples

TABLE I. RECOGNITION ACCURACY FOR ALL TESTED SAMPLES

Recognition Accuracy [%]	Recognized classes		
	 *Class1*	 *Class2*	 *Class3*
Case 1-*dH/dg*	5/**84.8%**	7/ **78.8%**	8/ **75.7%**
Case 2-*Exp(100.H(g))*	3/ **90.9%**	5/ **84.8%**	6/ **81.8%**
Case 3-alternately (dH/dg; Exp(100.H(g))	0/ **100%**	2/ **94%**	1/ **97%**

layer neurons and 3 output neurons, corresponding to the three trained classes. Figures 4 through 7 represent respectively H(g), train dH(g)/dg, train Exp(100.H(g)) and test Exp(100.H(g)) values for four samples of each class. The achieved output neuron values when recognizing samples of class 1, 2 and 3 are shown in Figures 8, 9 and 10. The proportion of 60%-7%-33%: 60 train samples, 7 verification samples, 33 test samples of each class) between training, cross validating and testing set of the general sample number is used in the research [11]. The 60% of the samples for each class were randomly given to the MLP NN for training with 20 samples of each class. To some of the train exemplars Motion Blur or Gaussian Noise is added. Motion Blur is added to simulate the effect of smoothing and blurring the images, when they move on a conveyer belt. The value of 9Pix Motion Blur corresponds to an image resolution of 300 dpi or 118 Pix/cm, to 25 m/min linear velocity of the conveyer belt and to 1/500 sec camera exposure time. The same conditions but for 1/300 exposure time correspond to 15Pix Motion Blur and for 1/200 exposure time corresponds to 25 Pix Motion Blur. Gaussian Noise 2%, 3% or 9Pix Motion Blur to three of the train samples of each class was added. To five of the test samples for each class was added Gaussian Noise between 3 and 5% or Motion Blur between 10 and 15%.

The train process terminated when a Mean Square Error (MSE) of 0.01 was obtained. The recognition accuracy is calculated as (1 - Number of false recognized samples/Number of all test samples of each class) x 100 [%] and is given in Table I. The results are given for three different training modes: first case - train the NN only with dH(g)/dg; second case – train only with Exp(100.H(g)); third – train alternatively with both dH(g)/dg and Exp(100.H(g)). The best recognition accuracy between 94% and 100% is obtained in the third case. The output results are extracted through *VB for AI* in different conventional interface formats as Modbus, RS 232 and GigE Vision Standard. Table II shows the comparative results concerning recognition accuracy and real-time execution. They are related to the research given in [5][6] where the same images were tested, but applying pre-processing with Wavelets (DWT) and DCT over grey image histograms. Almost the same recognition accuracy was achieved as with DWT, but with a simplified

TABLE II. COMPARATIVE RESULTS FOR RECOGNITION ACCURACY AND REAL-TIME EXECUTION

Method	Number of hidden neurons	MSE [%]	Recognition accuracy [%]	Real-time execution [ms]
Histogram	50	0.16	85	578
DCT	50	0.01	95	638
DWT	25	0.16	100	649
Alternately (dH/dg; Exp(100.H(g))	18	0.01	97	247

NN structure (only 18 neurons in the hidden layer) because of simple pre-processing method providing at the same time de-correlation of the NN input train data. In the case of alternately training with dH/dg; Exp(100.H(g), the execution time is about three times reduced.

V. CONCLUSION

In this research, a simple method for recognition of similar marble tiles with high correlated histograms is proposed and tested for three texture classes. High recognition accuracy is obtained under very simple calculations in the preprocessing stage. Calculation of dH(g)/dg and Exp(100.H(g)) is a very simple single operation over H(g). Training the MLP NN with both – slightly de-correlated inter class data as dH(g)/dg, thus conserving the local changes of H(g) between neighbors g, and strong de-correlated data as Exp(100.H(g)) is a prerequisit to obtain very good recognition results. The choice of only one NN with a relatively small number of neurons, instead of a hierarchical NN structure and the simple processing, allows method implementation in real-time systems. In future work, the method will be tested for more classes with similar textures also for other type of textures, to generalize the results. It is also interesting to find analog transformations for good NN input data de-correlation.

REFERENCES

[1] C. W. D. Almeida, R.M.C.R. de Souza, and A. L. B. Candeias, "Texture Classification Based on a Co-Occurrence Matrix and Self-Organizing Map," IEEE International Conference on Systems Man & Cybernetics, University of Pernambuco, Recife, pp. 2487-2491, 2010.

[2] G. A. Azim, and S. Nasir, "Textile Defects Identification Based on NNs and Mutual Information," International Conference on Computer Applications Technology (ICCAT), Sousse Tunisia, pp. 1-8, 2013.

[3] J. M. C. de-V. Alajarin, T. Balibrea, and M. Luis, "Marble Slabs Quality Classifcation System using Texture Recognition and NNs Methodology," ESANN'1999 proceedings - European Symposium on Artificial NNs, Bruges, Belgium, pp. 75-80, 1999.

[4] D. Feng, Z. Yang, and X. Qiao, "Texture Image Segmentation Based on Improved Wavelet NN," LNCS, Springer, Heidelberg, vol. 4493, pp. 869–876, 2007.

[5] I. Topalova, "Automated Marble Plate Classification System Based on Different NN Input Training Sets and PLC Implementation," IJARAI – International Journal of Advanced Research in Artificial Intelligence, Volume1, Issue2, pp. 50-56, 2012.

[6] I. Topalova, “Recognition of Similar Wooden Surfaces with a Hierarchical NN Structure,” SAI/ IJARAI – International Journal of Advanced Research in Artificial Intelligence, Volume 4, Issue10, pp. 35-39, 2015.

[7] B. Widrow and S. Stearns, “Adaptive Signal Processing,” Prentice-Hall, Inc. Englewood Cliffs, N.J. 07632, pp.36-40, 2004.

[8] R. C. Gonzalez, and R. E Woods, “Digital Image Processing”, 3rd Edition, Prentice Hall, India, 2008.

[9] Weisstein, Eric W. "Correlation Coefficient." From MathWorld, A Wolfram Web Resource. Available from: http://mathworld.wolfram.com/CorrelationCoefficient.html, 2017.

[10] Vision Builder AI, User Manuel, Copyright © 2013, pp. 45-58, 2013.

[11] Neuro Solutions, Copyright © 2014 by NeuroDimension, pp. 67-79, 2015.