

# Recognition of Similar Marble Textures through Different Neural Networks with De-correlated Input Data

## Short Paper

Irina Topalova

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

Magdalena 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 at high recognition accuracy, combined with simple pre-processing, the neural network is trained with different alternating input training sets including combination of high correlated and de-correlated input data. The de-correlated input data are also used for training of a self-organized map neural network, aiming to prove the efficiency of the pre-processing method also for unsupervised neural networks. The obtained good results in the recognition stage are represented, compared and discussed. Further research is proposed.

**Keywords**- MLP neural network; SOM neural network; texture recognition; pre-processing; de-correlation

### I. INTRODUCTION

This article is a continuation of the study by the same authors and published at the conference “IARIA/ICAS’2017”, Barcelona, Spain [1]. The automated recognition of marble slab surfaces is an important 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 pre-processing 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 pre-processing, 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 pre-processing 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 [2], 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 [3], propose also a method based on a GLCM feature extractor. The numerical simulation shows error recognition of 91%. The authors in [4], 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 [5][6]. Training a hierarchical NN

structure with texture histograms and their second derivative is also announced as giving good recognition accuracy [7]. Recently, the authors of [8] have published a color balancing model for texture recognition and implementation of convolutional neural networks (CNN). Their approach includes texture images acquired under several different lighting conditions. Since the neural network can be trained inefficiently when the training set is not big enough, some authors offer appropriate variations in the learning stage in order to obtain good recognition results [9]. These authors offer an alternative to the full training procedure, adapting an already trained network to a new classification, by additional training only a chosen subset of parameters. The authors of [10] offer color texture descriptors that measure local contrast. These descriptors are less sensitive than the colors themselves to variations in illuminance. The same authors enhanced their method by proposing a novel colour space where changes in illumination are even simplified [11]. J. M. P. Batista presents a method for classification of color marble textures, using logistic regression, first order fuzzy Takagi-Sugeno system, based on the clustering algorithms and Fuzzy C-Means [12].

Considering the explicated data, we could formulate some disadvantages of the approaches given above. The obtained accuracy of 97.8% in [2] is only for textures that are not very similar, i.e. they are not overlapping in the parametrical feature space. The use of GLCM needs high computations and even faster version of a Co-occurrence matrix as given in [4], needs computations 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 [7], with different input feature vectors, would be more complicated, particularly for real-time applications in different hardware platforms. The obtained accuracy is high, but not approaching 100%. The authors of [8] apply a complex approach without taking into account that different lightening for the same textures results in the translation of the histogram of the image along the X axis, without substantially altering its shape. If the translated histogram is used as an input vector on a suitable neural network, it will be able to make a translational invariant recognition. Changes in the texture histogram and along the Y axis have to be taken into account as they are influenced by the contrast changes between the local segments. In this way the algorithm would be greatly simplified. The approach given in [9] requires additional training by choosing appropriate subset of texture parameters, which would complicate the algorithm. The authors of [10][11] propose simplified descriptors that are less sensitive to variations in illuminance, but it still requires significant computing resources. The study given in [12] applies a complex method of recognizing marble textures but achieves a relatively low accuracy of 83.54% and there is a need to speed up the algorithm, because it gives 1.3 sec per marble texture.

Thus, the important source of optimization for the recognition method lies in a simplification of the pre-processing stage /the input feature vector and in finding a Method and System Development more efficient training

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

#### A. Selecting a Pre-processing Method

Complying with the finding that NN training would be more efficient, when applying different types of intra class input data [13][14], we choose to training 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,  $U = \text{Exp}(k.H(g))$  gave the best reduction of the inter class correlation coefficient. It was chosen for second input training set. So, the MLP NN is trained with these alternating input training sets including combination of high correlated and de-correlated input data. The de-correlated input data are also used for learning of a SOM neural network, aiming to prove the efficiency of the pre-processing method also for unsupervised neural networks, verifying the good impact of the de-correlated input data on the training facility.

#### B. System Components

The proposed test system includes one smart camera *NI 1742(300dpi)* with triggered infrared lighting, software Vision Builder for Automated inspection [15] *AI'14 (VB for AI)* and Neuro-System V5.0 - shown in Figure 1. The images are taken at the same distance with the same spatial resolution. The system works in two modes - off-line or training and on-line, or recognition and classification. In both modes, first the contrast quality for the captured images is improved in *VB for AI*, applying simple lookup logarithmic power square function, followed by the corresponding pre-processing of the two types of training sets. In off-line /or training mode/, the two types of calculated training sets of all classes are applied to the inputs of the proposed neural network structures (MLP or SOM). The training process ends with the result - two matrices of weighting coefficients  $\mathbf{W}_{\text{MLP}}$  and  $\mathbf{W}_{\text{SOM}}$ . In on-line /or test mode - recognition and classification/ the same operations are performed for each test sample, but the input data only “go” through the saved (after the training), weight matrixes  $\mathbf{W}_{\text{MLP}}$  or  $\mathbf{W}_{\text{SOM}}$ . The results are given to *VB for AI* for visualization and preparation for extraction through standard interfaces.

### III. EXPERIMENTS AND RESULTS

In this section, the details of the pre-processing stage are given, along with a description of the MLP NN and of the SOM 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. The pre-processing stage is presented in subsection A, the MLP and

SOM NN's training methods are explained in subsections B and C respectively.

**A. Pre-processing Stage**

The experiments are carried out for nine marble tiles/classes with similar textures given in Figure 3. 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 and SOM NN, would require a much more complex NN structures.

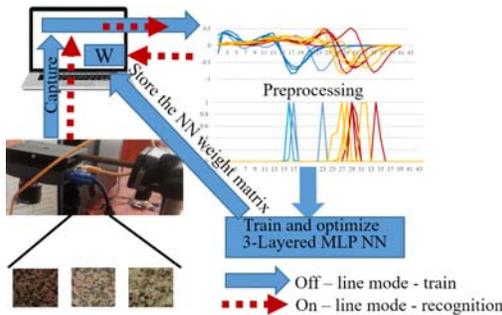


Figure 1. System components

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

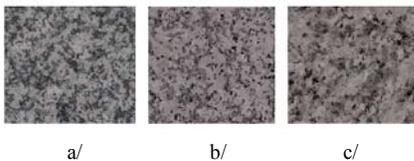


Figure 2. Grey level marble tiles – a/-class1, b/-class2, c/-class3

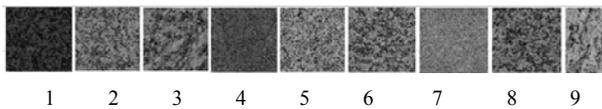


Figure 3. All tested classes of similar marble textures

classes  $i, j$  for different input NN feature vector descriptions, the correlation coefficient  $r_{ij}$  is calculated according to [16]. Points 1 to 4 of X axis in Figure 4 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, shown in Figure 3. As the coefficient  $r_{ij}$  for  $H(g)$  varies in the range  $(-0.24;0.96)$ , it shows very high similarity 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  $k \in \mathbf{R}$ ,

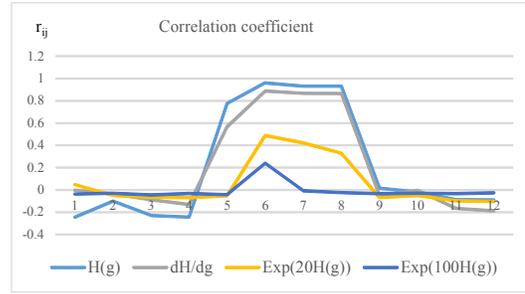


Figure 4. Correlation coefficient  $r_{ij}$  for different input training sets

will be suitable. We choose this function 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 4, where  $r_{ij}$  varies in the range  $(-0.036;0.24)$ . For the normalized  $H(g)$  values given in Figure 5, the calculated  $U$  are represented in Figure 6. As the function  $U$  has a smoothing effect over  $H(g)$ , it also 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  $r_{ij}$  than  $H(g)$ . The training set of  $dH(g)/dg$  is shown in Figure 7.

**B. Training Method for MLP NN**

The decision plane consists of a 3-layered MLP NN, trained with well-known Backpropagation algorithm [17]. The input layer is connected with 45  $dH(g)/dg$  and  $U=Exp(100.H(g))$  sampled values over the histograms, according to the requirements for signal/histogram reconstruction, proved by Shannon sampling theorem [18]. This sampling allows a reduction of the input vector. Both types of vectors are applied 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

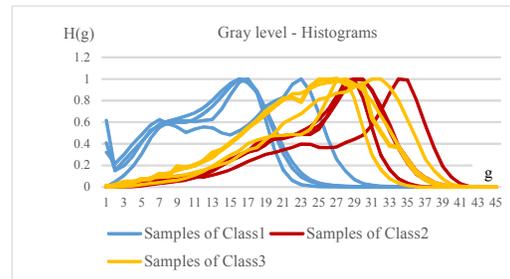


Figure 5. Normalized histogram values  $H(g)$  for samples of classes 1, 2, 3

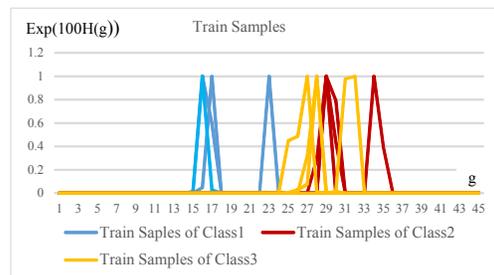


Figure 6. Training  $Exp(100.H(g))$  values for the samples of classes 1, 2, 3

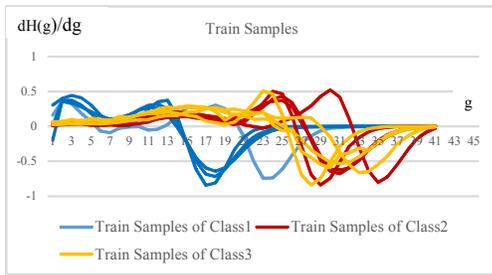


Figure 7. Training  $dH(g)/dg$  values for the samples of classes 1, 2 and 3

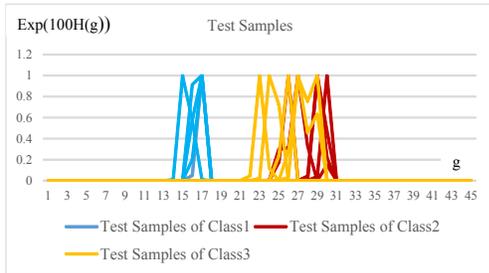


Figure 8. Test  $Exp(100.H(g))$  values for the samples of classes 1, 2 and 3

near to "1" or "-1" (using tangent hyperbolic as activation function). The training in off-line mode was repeated to find the optimized MLP NN structure according to the method given in [6]. We obtained the best fitting structure with 18 hidden layer neurons and 3 output neurons, corresponding to the three trained classes. Figures 5 through 8 represent respectively  $H(g)$ , training  $dH(g)/dg$ , training  $Exp(100.H(g))$  and test  $Exp(100.H(g))$  values for four samples of each class. The achieved output neuron values when recognizing samples of classes 1, 2 and 3 are shown in Figures 9, 10 and 11. Figure 12 shows the output neuron values for recognition of all test samples of the three classes. The proportion of 60%-7%-33%: (60 training 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 [17]. 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 training exemplars Motion Blur or Gaussian Noise is added. Motion Blur is added to simulate the effect of smoothing and blurring the images, when they are moving on a conveyor 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 conveyor 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 training 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 training process terminated when a Mean Square Error (MSE) of 0.01 was obtained. The recognition accuracy is calculated as  $(1 - \text{Number of false recognized samples}/\text{Number of all test$

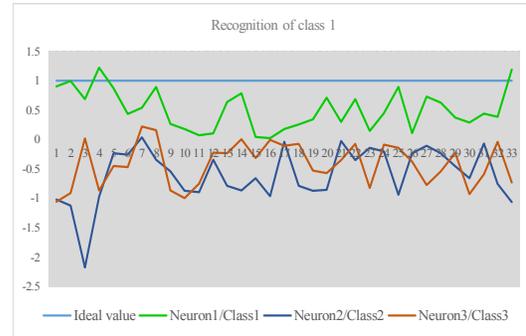


Figure 9. Output neuron values for recognition of class 1 samples

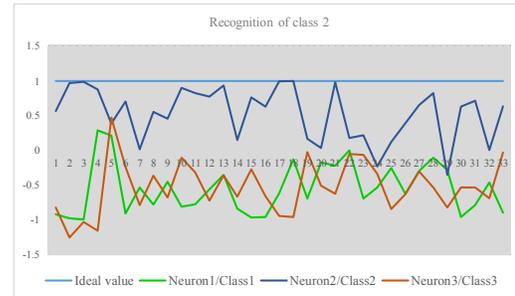


Figure 10. Output neuron values for recognition of class 2 samples

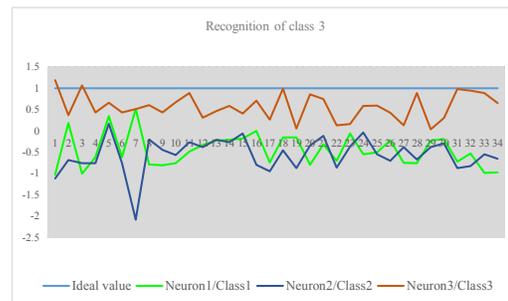


Figure 11. Output neuron values for recognition of class 3 samples

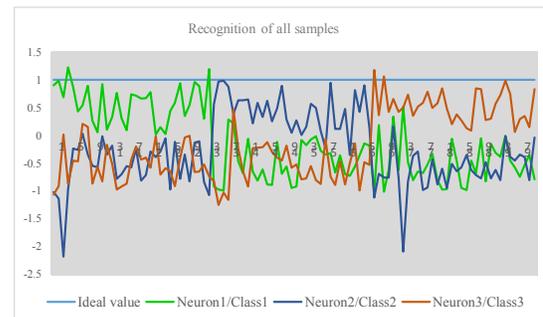


Figure 12. Output neuron values for recognition of all 33 test samples of the three classes

samples of each class)  $\times 100$  [%] and is given in Table I. The results are given for three different training modes: first case - training the NN only with  $dH(g)/dg$ ; second case - training only with  $Exp(100.H(g))$ ; third - training alternatively with both  $dH(g)/dg$  and  $Exp(100.H(g))$ . The best recognition

TABLE I. RECOGNITION ACCURACY FOR ALL TESTED SAMPLES

Recognition Accuracy [%]	Recognized classes								
	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8	Class9
Case 1-dH/dg	5/84.8%	7/ 78.8%	8/ 75.7%	5/84.8%	8/ 75.7%	8/ 75.7%	5/84.8%	7/ 78.8%	6/ 81.8%
Case 2-Exp(100.H(g))	3/ 90.9%	6/ 81.8%	6/ 81.8%	3/ 90.9%	5/ 84.8%	6/ 81.8%	4/ 87.8%	5/ 84.8%	3/ 90.9%
Case 3-MLP-alternately (dH/dg; Exp(100.H(g))	1/ 97%	2/ 94%	1/ 97%	1/ 97%	3/ 90.9%	3/ 90.9%	1/ 97%	2/ 94%	1/ 97%
Case 4-SOM200 Exp(100.H(g))	0/ 100%	2/ 94%	2/ 94%	0/ 100%	1/ 97%	1/ 97%	0/ 100%	2/ 94%	0/ 100%

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 [6][7] 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 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 training data. In the case of alternately training with dH/dg; Exp(100.H(g)), the execution time is about three times reduced.

C. Training Method for SOM NN

To prove the efficiency of the pre-processing method also for unsupervised neural networks, verifying the good impact of the de-correlated input data on the training facility, a SOM NN is trained with the same and only with the U=Exp(100.H(g)) values. Here we apply the Kohonen

SOM algorithm [19] with a topology shown in Figure 13. We use 45 Input neurons and different number of SOM neurons, in series with 4x4, 7x7 and finally with 20x10 neurons. The size of the SOM grid is determined empirically, considering the recommendations given by the Kohonen himself [19]. It stands that the size of the SOM grid array must roughly correspond to the major dimension of the distribution of the Input data [20]. As the reducing the real-time execution is desirable when applying the methods for real-time applications we begin the training with a small size of SOM grid (4x4) and increase this number (7x7) until good recognition accuracy in the test phase is achieved (20x10). We choose a hexagonal SOM grid. These network was trained with initial learning rate of 0.06, initial neighbourhood size of 200 and neighbourhood decay amount of 0.5. Figure 13 represents the test results for the 9 classes. It is visible that the small grid gives bad results with high overlapping of recognized samples, but increasing the size to 20x10 neurons gives very good clustering. In the best case only two samples of class 2 overlap with one sample of class 5 and class 6. Also one sample of class 3 overlaps with two samples of class 8. When calculating the recognition accuracy over the whole number of test samples of all 9 classes, i.e. 9x33=297, with only 8 false clustered samples, it gives 97.3% accuracy. Table I and Table II reflect the obtained recognition accuracy and execution time also for the tested SOM NN. It is distinct that the SOM gives also very high accuracy along with simple pre-processing and in

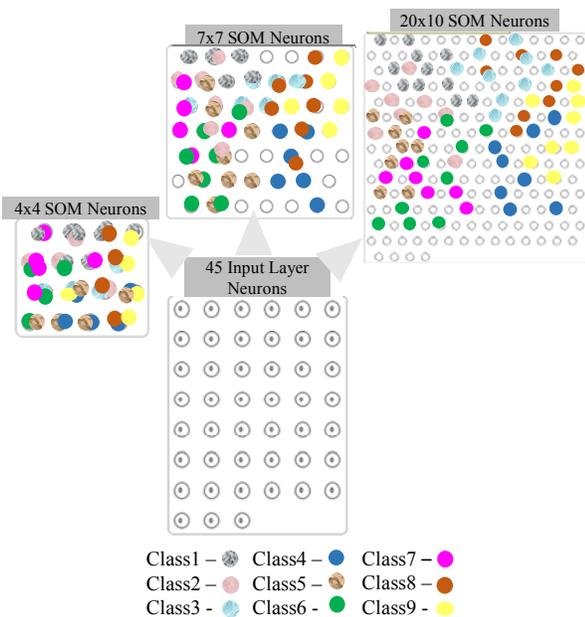


Figure 13. Test results in SOM NN structure with different grid size

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

Method	Number of hidden neurons	MSE [%] / Learning Rate for SOM	Recognition accuracy [%]	Real-time execution [ms]
MLP-Histogram	50	0.16	85	578
MLP-DCT	50	0.01	95	638
MLP-DWT	25	0.16	100	649
MLP-alternately (dH/dg; Exp(100.H(g))	18	0.01	97-100	247
SOM-Exp(100.H(g))	200 SOM Neurons	1.23E-321	94-100	112

addition gives shorter execution time in comparison to MLP NN. The most recent results obtained by the authors in [12] are 1.3 sec per marble texture, with relatively low recognition accuracy of 83.54%. Comparing the results achieved in terms of computing performance and accuracy, we could say that the presented method offers significantly better performance.

#### IV. CONCLUSION

In this research, a simple method for recognition of similar marble tiles with high correlated histograms is proposed and tested for nine texture classes. High recognition accuracy is obtained under very simple calculations in the pre-processing stage. Calculation of  $dH(g)/dg$  and  $\text{Exp}(100.H(g))$  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  $\text{Exp}(100.H(g))$  is a prerequisite to obtain very good recognition results and makes it possible to implement this method in different real-system systems. 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. It is also interesting to find analog transformations for good NN input data de-correlation. The achievement of high recognition accuracy in shorter execution time for SOM NN, by the same de-correlated input data proves the generalization of the proposed method. It is also interesting to find analog transformations for good NN input data de-correlation.

In future work, the method will be tested for more classes with similar textures also for other type of textures, to generalize the results. For example, the study can also be applied to similar textures on wooden surfaces. Another interesting idea for us is to first apply only SOM NN, to group / categorize in advance the proposed de-correlated data, after which the values of SOM neurons are submitted as inputs to the MLP network. In this way, it would be possible to precisely distinguish small local variations in textures, such as minor defects.

#### REFERENCES

- [1] I. Topalova and M. Uzunova, "Neural Network Structure with Alternating Input Training Sets for Recognition of Marble Surfaces," IARIA/ ICAS'2017 – the Thirteenth International Conference on Autonomic and Autonomous Systems, ISBN: 978-1-61208-555-5, May, pp.40-44, Barcelona, Spain, 2017.
- [2] 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.
- [3] 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.
- [4] J. M. C. de-V. Alajarin, T. Balibrea, and M. Luis, "Marble Slabs Quality Classification System using Texture Recognition and NNs Methodology," ESANN'1999 proceedings - European Symposium on Artificial NNs, Bruges, Belgium, pp. 75-80, 1999.
- [5] D. Feng, Z. Yang, and X. Qiao, "Texture Image Segmentation Based on Improved Wavelet NN," LNCS, Springer, Heidelberg, vol. 4493, pp. 869–876, 2007.
- [6] 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.
- [7] 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.
- [8] S. Bianco, Cl. Custano, P. Napolitano, and R. Schettini, "Improving CNN-Based Texture Classification by Color Balancing," Journal of Imaging, 27 July, 2017.
- [9] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition", Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, USA, 23–28 June, 2014.
- [10] C. Cusano, P. Napolitano, and R. Schettini, "Combining local binary patterns and local color contrast for texture classification under varying illumination," J. Opt. Soc. Am. A, 31, 1453–1461, 2014.
- [11] C. Cusano, P. Napolitano, and R. Schettini, "Local Angular Patterns for Color Texture Classification," New Trends in Image Analysis and Processing – ICIAP 2015 Workshops. Murino, V., Puppo, E., Sona, D., Cristani, M., Sansone, C., Eds.; Springer International Publishing: Cham, Switzerland, pp. 111–118, 2015.
- [12] J. M. P. Batista, "Marble Polished Stones Automatic Classification," Universidade de Lisboa, Portugal, November, 2015. [https://fenix.tecnico.ulisboa.pt/downloadFile/1126295043834456/Resumo\\_Alargado.pdf](https://fenix.tecnico.ulisboa.pt/downloadFile/1126295043834456/Resumo_Alargado.pdf), last access – 20.11.2017.
- [13] B. Widrow and S. Stearns, "Adaptive Signal Processing," Prentice-Hall, Inc. Englewood Cliffs, N.J. 07632, pp.36-40, 2004.
- [14] R. C. Gonzalez and R. E Woods, "Digital Image Processing," 3rd Edition, Prentice Hall, India, 2008.
- [15] Vision Builder AI, User Manuel, Copyright © 2013, pp. 45-58, 2013.
- [16] E. W. Weisstein, "Correlation Coefficient," From MathWorld, A Wolfram Web Resource. Available from: <http://mathworld.wolfram.com/CorrelationCoefficient.html>, 2017.
- [17] Neuro Solutions, Copyright © 2014, NeuroDimension, pp. 67-79, 2015.
- [18] St. W. Smith, "The Scientists and Engineer's Guide to Digital Signal Processing," Book, Copyright © 1997-2011 by California Technical Publishing, 2011.
- [19] Sh. M. Guthikonda, "Kohonen Self-Organizing Maps," Wittenberg University, pp.8-10, December, 2005.
- [20] T. Kohonen and T. Honkela, "Kohonen network," Scholarpedia. Retrieved 2012-09-24, 2012.