A Preprocessed Counterpropagation Neural Network Classifier for Automated Textile Defect Classification

Mokhlesur Rahman
Department of Computer Science and Engineering, Prime University, Dhaka, Bangladesh
Email: mmarks_cse@yahoo.com

Tarek Habib
Department of Computer Science and Engineering, Green University of Bangladesh, Dhaka, Bangladesh
Email: md.tarekhabib@yahoo.com

***Abstract*—Counter Propagation Neural Network (CPN) is a hybrid neural network because it makes use of the advantages of supervised and unsupervised training methodologies. CPN has a reputation for high accuracy and short training time. In this paper, a variant of CPN, namely preprocessed Counter Propagation Neural Network is proposed. We propose that if some preprocessing can be introduced to assign weights instead of random weight assignment during CPN training, it will result in good classification accuracy, very short training time and simple model complexity. The preprocessed CPN has promising applicability in a number of domains, among which textile defect classification is a prominent one. Textile sector is the most prospective export sector in Bangladesh. We demonstrate the utility and capability of our preprocessed CPN classifier in automated textile defect classification in the context of Bangladesh. We have found very good results.**

***Index Terms*—Textile Defect, Automated Inspection, Defect Classification, Counterpropagation Neural Network (CPN), Preprocessed CPN, Centroid, Accuracy**

I. Introduction

An Artificial Neural Network (ANN) is a powerful classification tool that has had profound impact on the many most recent developments in scientific research. ANNs are suitable enough for real-time systems because of their parallel-processing capability. Counterpropagation Neural Network (CPN) classifier, which is capable of handling complex classification problems with good classification accuracy, has been investigated in different application domains, e.g. on-line handwritten character recognition [1], trademark recognition [2], face detection and recognition [3], and textile defect classification [4]. In this paper, we propose a variant of CPN, named preprocessed CPN (PCPN), which is capable of handling complex classification problems with good classification accuracy, very short training time and simple model complexity. We show that some preprocessing can be introduced to produce weights, which is assigned instead of random weights during network training. Assigning pre-calculated weights makes the training time much shorter and model complexity simple by retaining good classification accuracy. So this PCPN emerges as a very good choice of a classifier in order to address the problem of textile defect classification.

The Ready–Made Garments (RMG) is the major prospective export sector in Bangladesh and so is in the many other developing countries in the world. The RMG sector of Bangladesh has appeared as the largest earner of foreign currency, which has been growing exponentially since the 1980s. The amount of export of RMG sector of Bangladesh was 24491.88 million US$, which was 81.13% of total export of Bangladesh in the fiscal year 2013-14 [5]. Though this sector is not so efficient in industrial processes, it is trying to catch up with the rest of the world and attain the consumers' maximum satisfaction. So textile industry should improve quality of the production process for increasing current level of performance in the highly competitive global market. An important aspect of quality improvement is wastage reduction through accurate and early stage detection of defects in fabrics, which can be ensured by introducing automated fabric inspection in the place of manual inspection. Two difficult problems are mainly posed by automated fabric defect inspection systems. They are defect detection and defect classification. Since automated fabric defect inspection system is a real-time system, ANN is a very good choice for defect classification. Although applicability of CPN on textile defect classification has been thoroughly investigated in [4], PCPN can be applied in this context for increasing performance. Likewise it can also be applied on the domains of on-line handwritten character recognition, trademark recognition, face detection and recognition and so on.

This paper is organized as follows. Section II describes current state of solution to address the problem of fabric defect classification and Section III describes the

Manuscript received July 1, 2015; revised May 1, 2016.

doi: 10.18178/jiii.4.3.209-217

proposed PCPN model along with algorithm. In Section IV, we describe our approach to solve the problem. Section V describes how we implement our PCPN model and the results obtained after implementation. In Section VI, we have compared the results of PCPN with automated textile defect classification results in order to develop an understanding of the merits of our PCPN model. Finally, we conclude along with the limitations of our work and the scope and opportunity for future work in Section VII.

II. Background and Related Works

A large number of efforts have been given for automated textile defect inspection [4], [6]-[25]. Most of them have focused on defect detection, where some of them have given attention to classification. ANNs have been used as classifiers in a number of articles. Different learning algorithms have been used in order to train the ANNs.

Habib and Rokonuzzaman [4] have used CPN for textile defect classification. They focused on classifying textile defects using CPN model. They have performed through investigation on interrelationship between design parameters and performance of CPN model. Again, Habib and Rokonuzzaman [6] have deployed CPN in order to classify four types of defects. Basically, they concentrated on feature selection rather than giving attention to the CPN model. They have not performed in-depth investigation on interrelationship between design parameters and performance of backpropagation model.

Backpropagation learning algorithm has been used in [7]-[12]. Habib and Rokonuzzaman [7] have emphasized on classifying textile defects using backpropagation model. They have performed through investigation on interrelationship between design parameters and performance of backpropagation model. Again, Habib and Rokonuzzaman have focused on feature selection rather than focusing on the ANN model in [8]. They have used four types of defects and two types of features. Saeidi *et al*. [9] have first performed off-line experiments and then performed on-line implementation. In both cases, they have used six types of defects. Karayiannis *et al*. [10] have used seven types of defect. They have used statistical texture features. Kuo and Lee [11] have used four types of defect. Mitropulos *et al*. [12] have used seven types of defects in their research. Detailed investigation on interrelationship between design parameters and performance of ANN model has not been performed in any of these works discussed.

Resilient backpropagation learning algorithm has been used to train ANN in [13], [14]. They have worked with several types of defects considering two of them as major types and all other types of defects as a single major type. They have not reported anything detailed regarding the investigation of finding an appropriate ANN model.

Shady *et al*. [15] have used Learning Vector Quantization (LVQ) algorithm in order to train their ANNs. They have used six types of defects. They have separately worked on both spatial and frequency domains for defect detection. Kumar [18] has used two ANNs separately. The first one was trained by backpropagation algorithm. He has shown that the inspection system with this network is not cost-effective. So he has further used linear ANN trained by least mean square error (LMS) algorithm. The inspection system with this ANN is cost-effective. Karras *et al*. [20] have also separately used two ANNs. They have trained one ANN by backpropagation algorithm and the other one by Kohonen's Self-Organizing Feature Map (SOFM). Thorough investigation on interrelationship between design parameters and performance of ANN model has not been reported in any of these reviewed works.

Habib, Faisal and Rokonuzzaman [22] have performed classification of textile defects using ANN model trained by genetic algorithm. They used four types of defects. Basically, they focused on investigating the feasibility of genetic algorithm for defect classification. They have not performed thorough investigation on interrelationship between design parameters and performance of genetic algorithm classifier.

Furferi *et al*. [24] have used Levemberg-Marquardt algorithm, a variant of backpropagation learning algorithm, in order to train their ANN for the grading of car seat fabric quality. They have used five quality classes. Furferi and Governi [25] have used the combination of a statistical method of a SOFM and a feed forward backpropagation ANN based approach to correctly classify woollen clothes to be recycled.

III. Proposed Preprocessed CPN Model

CPN was developed by Robert Hecht-Nielsen. It is a hybrid network that has combined an unsupervised Kohonen layer with a teachable output layer. It tries to minimize the number of processing elements and training time [26], [27].

A. Existing Algorithm

CPN is a winner-take-all competitive learning network. CPN has three layers, namely input, hidden and output as shown in Fig. 1. The input, hidden and output layers are called input buffer layer, Kohonen layer and Grossberg layer, respectively. The learning process of the network consists of 2 (two) phases, namely Kohonen unsupervised learning phase and Grossberg supervised learning phase.

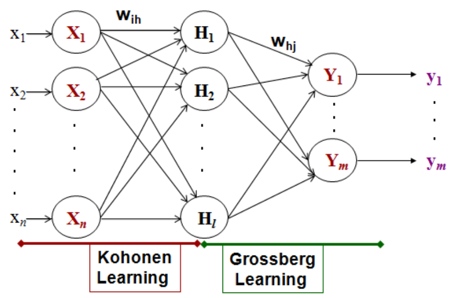


Figure 1. The architecture of CPN.

B. Proposed Algorithm

Our proposed algorithm works on the architecture same as the architecture of existing CPN. Some preprocessing is introduced before the data is entered into the CPN. We are given a training set $\{(X_1, T_1), (X_2, T_2), \ldots, (X_p, T_p)\}$ consisting of p ordered pairs of n- and m-dimensional vectors, which are the input and output vectors, respectively. Suppose we want to classify data into m classes. We need to compute the centroids of classes ($c_1, c_2, c_3, \ldots, c_m$) as the preprocessing part. We use them in assigning weights instead of assigning weights randomly in the Phase I of learning process. The idea is that the computing unit in hidden layer, H_h, will represent the class h and the centroid of that class as assigned weight will attract the vectors of the same class quickly and powerfully. This will result in very short training time. Moreover, the number of computing units in the hidden layer need not big in this approach. The entire algorithm is shown in Fig. 2 as pseudocode.

Phase I: (Kohonen unsupervised learning)

1. Compute all centroid vectors $c_h = (c_{h1}, c_{h2}, c_{h3}, \ldots, c_{hn})$, where $1 \le h \le m$.
2. (A) Assign centroid vector c_h to all the connections from all the nodes in the input layer to the h-th node in the hidden layer ($w_{1h}, w_{2h}, w_{3h}, \ldots, w_{nh}$), where $1 \le h \le m$.

   (B) Randomly assign value to each connection from the i-th node in the input layer to the h-th node in the hidden layer w_{ih}, where $m < h \le l$.

   (C) Randomly assign value to each connection from the h-th node in the hidden layer to the j-th node in the output layer node w_{hj}, where $1 \le h \le l$ and $1 \le j \le m$.
3. Computes the Euclidean distance between input vector and the weights of each hidden node.
4. Find the winner node with the shortest distance.
5. Adjust the weights that are connected to the winner node in hidden layer with $\Delta W_{ih^*} = \eta_K \cdot (X_i - W_{ih^*})$.

Phase II: (Grossberg Supervised Learning)

1. Same as (3) and (4) of phase I.
2. Let the link connected to the winner node to output node be set as 1 and the others be set to 0, i.e.

$$f(H_h) = \begin{cases} 1, & if\ h = h^* \\ 0, & otherwise \end{cases}.$$

3. Adjust the weights using $\Delta W_{h^*j} = \eta_G \cdot (T_j - Y_j) \cdot f(H_{h^*})$.

Figure 2. The preprocessed CPN algorithm.

IV. METHODOLOGY

We are to address the problem of empirically discovering such network design parameters, i.e. Kohonen learning constant (η_K), Grossberg learning constant (η_G) and model complexity (number of computing units in the hidden layer), for which the performance metrics, namely accuracy and training time, are optimized. Our intention is to maximize accuracy and minimize training time. Both accuracy and training time are dependent on η_K, η_G and model complexity. If we denote accuracy, training time, model complexity, the number of computing units in the input, hidden and output layer by A, T, C_M, N_I, N_H and H_O respectively, then

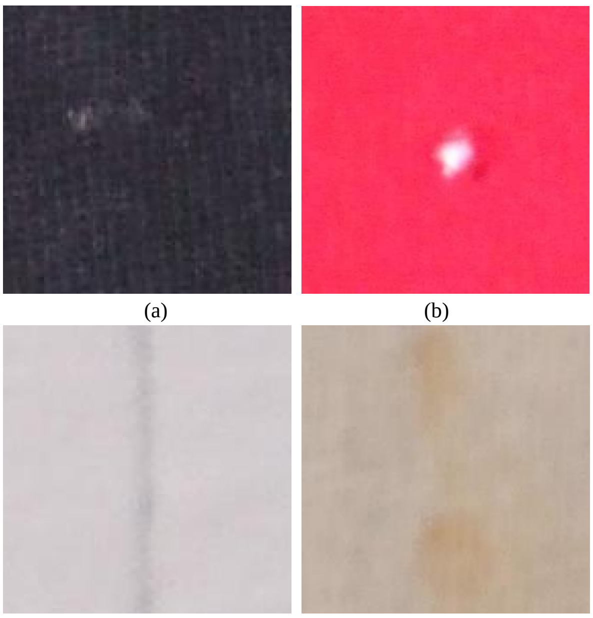
$$A = f_1(C_M, \eta_K, \eta_G)$$
$$\text{and } T = f_2(C_M, \eta_K, \eta_G),$$
$$\text{where } C_M = (N_I, N_H, N_O).$$

So the computational problem becomes as follows:

$$\text{maximize} f_1(C_M, \eta_K, \eta_G)$$
$$\text{and minimize } f_2(C_M, \eta_K, \eta_G)$$
$$\text{subject to } N_I = 4$$
$$N_H \geqslant 6$$
$$N_O = 6$$
$$0 < \eta_K < 1$$
$$0 < \eta_G < 1.$$

A. Types of Defects

In this paper, we have worked with four types of defects. These defects often appear in knitted fabrics in Bangladesh. They are color yarn, hole, missing yarn (vertical and horizontal) and spot. All of the defects are shown in Fig. 3.



(a) (b) (c) (d)

Figure 3. Different types of defects occurred in knitted fabrics. (a) Color yarn. (b) Hole. (c) Missing yarn. (d) Spot.

B. A Suboptimal Feature Set

Four features are selected out of seven originally available features by using sequential forward Selection

[28]. The selected four features comprise a suboptimal feature set in order to classify the defects successfully. Detailed information about all of the seven features can be found in [6] for interested readers.

C. Error Function

Error function (E) is chosen as the sum of squared error. A set of input vectors (X) and output/target vectors (T) is supplied to exhibit the desired network behavior. We are given a training set $\{(X_1, T_1), ..., (X_p, T_p)\}$ consisting of p ordered pairs of n- and m-dimensional vectors, which are the input and output vectors, respectively. When the input vector X_i from the training set is presented to this network, it produces an output Y_i different in general from the target T_i. Then the error function of the network is defined as follows:

$$E = \frac{1}{2}\sum_{i=1}^{p}\left\|Y_i\ T_i\right\|^2 \tag{1}$$

D. Tuning Network Design Parameters

We empirically discover the desired values of network design parameters, i.e. η_K, η_G and C_M (model complexity, i.e. number of computing units in the hidden layer) by tuning them. We tune η_K keeping the other two parameters unchanged. Then we tune η_G keeping the other two unchanged. Likewise we tune the two network design parameters η_K and η_G by changing their order. At last we tune C_M.

V. Implementation

We start with inspection images of knitted fabric of size 512×512 pixels, which are converted into a grayscale image. In order to smooth these images and remove noises, they are filtered by 7×7 low-pass filter convolution mask. Then gray-scale histograms of the images are formed. Two threshold values θ_L and θ_H are calculated from each of these histograms using histogram peak technique [29]. Using the two threshold values θ_L and θ_H, images with pixels $P(x, y)$ are converted to binary images with pixels $I_B(x, y)$, where

$$I_B(x,y) = \begin{cases} 1, & if\ \theta_L \le P(x,y) \le \theta_H \\ 0, & otherwise \end{cases} \tag{2}$$

These binary images contain objects (defects) if any exists, background (defect-free fabric), and some noises. These noises are smaller than the minimum defect wanted to detect. In our approach, we want to detect a defect of minimum size 3mm×1mm. So, any object smaller than minimum-defect size in pixels is eliminated from the binary images. If the minimum defect size in pixels is θ_{MD} and an object with pixels $Obj(x, y)$ is of size N_{obj} pixels, then

$$Obj(x,y) = \begin{cases} 1, & if\ N_{obj} \ge \theta_{MD} \\ 0, & otherwise \end{cases} \tag{3}$$

Then a number of features of defects are calculated, which forms feature vectors corresponding to defects present in images.

One hundred color images of defective and defect-free knitted fabrics of seven colors are acquired. So, the number of calculated feature or input vectors is 100. That means our sample consists of 100 feature vectors. Table I shows the frequency of each defect and defect-free class in our sample of 100 images.

TABLE I. Frequency of Each Defect and Defect-Free Class

No.	Class	Frequency
1	Color Yarn	7
2	Vertical Missing Yarn	16
3	Horizontal Missing Yarn	17
4	Hole	12
5	Spot	18
6	Defect-Free	30
	Total	100

The features provided by the feature extractor are of values of different ranges, which causes imbalance in feature space and makes the training phase difficult. The scaling, shown in (4), (5), (6), and (7), of the features is made in order to have proper balance in feature space, i.e. all feature values are calculated in 100. If H'_{DW}, W'_{DW}, $R'_{H/W}$ and N'_{DR} represent the scaled values of the features provided by the feature extractor H_{DW}, W_{DW}, $R_{H/W}$, and N_{DR}, respectively, then

$$H'_{DW} = \frac{H_{DW}}{512} \times 100 \tag{4}$$

$$W'_{DW} = \frac{W_{DW}}{512} \times 100 \tag{5}$$

$$R'_{H/W} = 100 \times R_{H/W} \tag{6}$$

$$N'_{DR} = \sqrt[500]{(N_{DR} - 1) \times 10^{999}} \tag{7}$$

We use the holdout method [30] for model evaluation. We split all feature vectors into two parts. One part consisting of 50 feature vectors is for training the PCPN model and the other part consisting of the rest 50 feature vectors is for testing. The target values are set to 1 and 0s for the corresponding class and the rest of the classes, respectively. That means if a feature vector is presented to the PCPN model during training, the corresponding computing unit in the output layer of the corresponding class of the feature vector is assumed to fire 1 and all other units in the output layer are assumed to fire 0.

We come up with a training data set $\{(X_1, T_1), (X_2, T_2),......., (X_{50}, T_{50})\}$ consisting of 50 ordered pairs of 4- and 6-dimensional vectors, which are the input and output vectors, respectively. We want to classify the data into 6 classes. We then compute the centroids of classes (c_1, c_2, c_3, ……, c_6) as the preprocessing part. We use them in assigning weights instead of assigning weights randomly in the Phase I of learning process. The computing unit in hidden layer, H_h, will represent the class h, where $1 \le h \le 6$. The PCPN model is trained with maximum number of training cycle 10^7, maximum amount of training time 1 hour and maximum tolerable error less than 10^{-4}. That means training continues until 10^7 training cycles and 1 hour is elapsed and error less than 10^{-4} is found. After the training phase is completed, the PCPN model is tested with all the feature vectors of the testing part. Then all feature vectors are again split into two parts. The PCPN model is preprocessed and trained with these new parts and then is tested. In this way, for a specific combination of network design parameters, the model is preprocessed, trained and tested twice since the training data are sampled as 50:50 ratio without replacement. We take the average of the two results.

In accordance with CPN architecture, we use three-layer feedforward ANN for our model. We started with a large CPN that has 4 computing units in the input layer, 10 computing units in the hidden layer and 6 computing units in the output layer. We describe the entire training in detail in the following parts of this section, i.e. Section V.

A. *Choosing Activation Function*

For our CPN, the unsupervised and supervised learning are the Kohonen and Grossberg learning respectively. For Kohonen unsupervised learning, we implement a piecewise activation function, which is defined as follows:

$$f(x)=\begin{cases}1, & \text{if } x \text{ is the closest according to} \\ & \text{the closeness criterion} \\ 0, & \text{otherwise}\end{cases} \qquad (8)$$

Here in (8), the closeness criterion is distance-based, i.e. the Euclidean distance between feature vectors and the weights of each computing unit in the hidden layer. For Grossberg supervised learning, we implement a linear activation function, which is defined as follows:

$$f(x)=x \qquad (9)$$

B. *Tuning η_K and η_G*

We first train as well as test the CPN for η_K = 0.01 and η_G = 0.01. We raise the value of η_K slowly, and train as well as test the ANN for that value of η_K keeping the value of η_G unchanged. The results thus obtained are shown in Table II and Fig. 4. We observe that E is tolerable, i.e. less than 10^{-4}, and the accuracy is maximum, i.e. 100%, for η_K = 0.3. Moreover, the number of elapsed training cycle is minimum, i.e. 215, for η_K = 0.3. So 0.3 is chosen as the value of η_K.

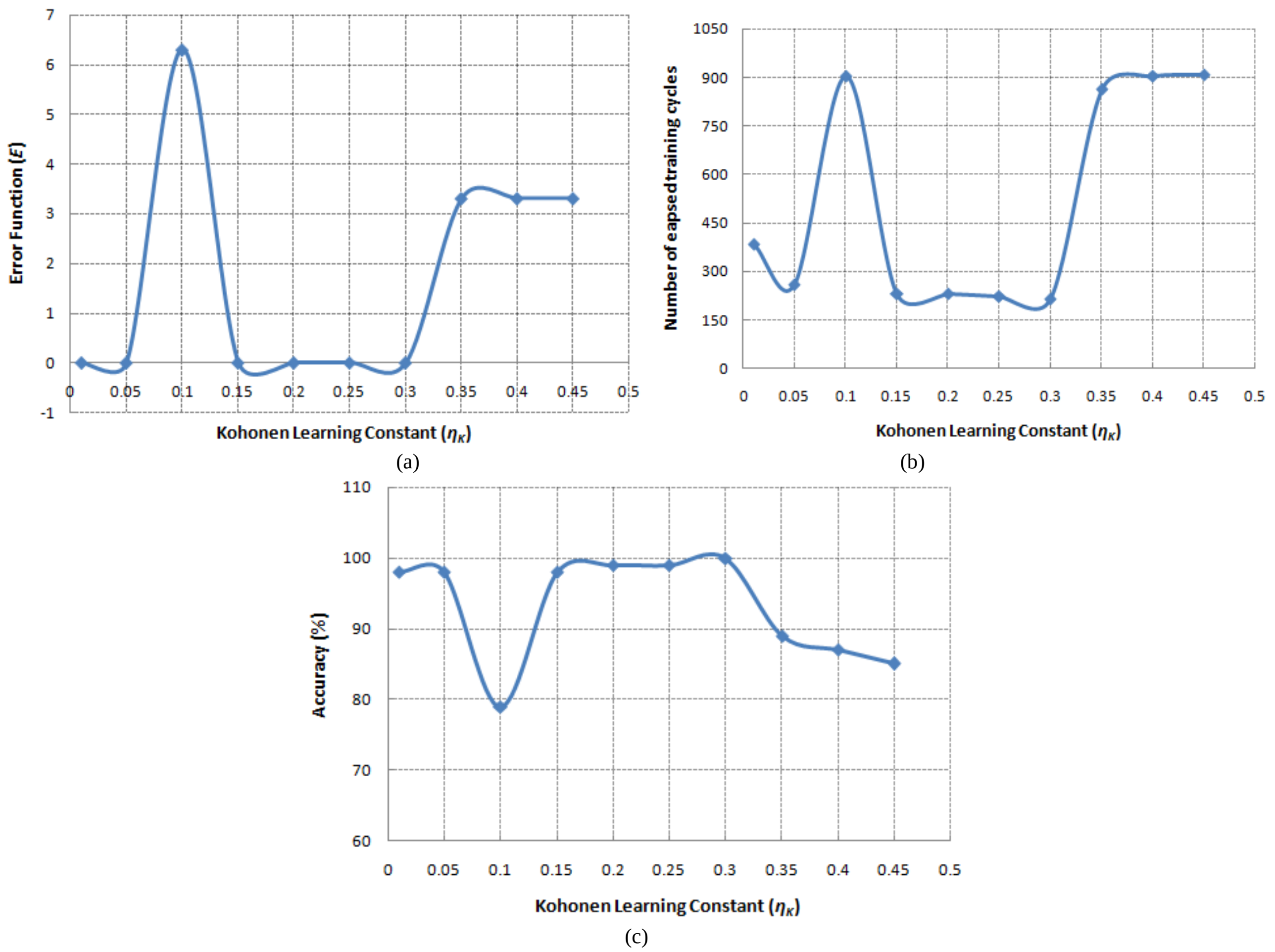


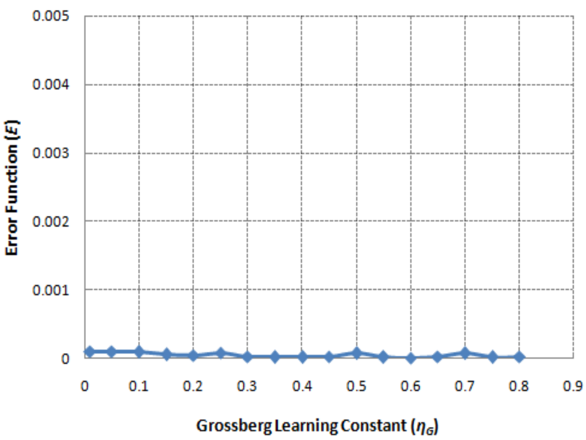
Figure 4. Results of tuning kohonen learning constant η_K keeping η_G fixed. (a) Error function (E). (b) Number of elapsed training cycle. (c) Accuracy.

TABLE II: RESULTS OF TUNING KOHONEN LEARNING CONSTANT H_K KEEPING H_G FIXED

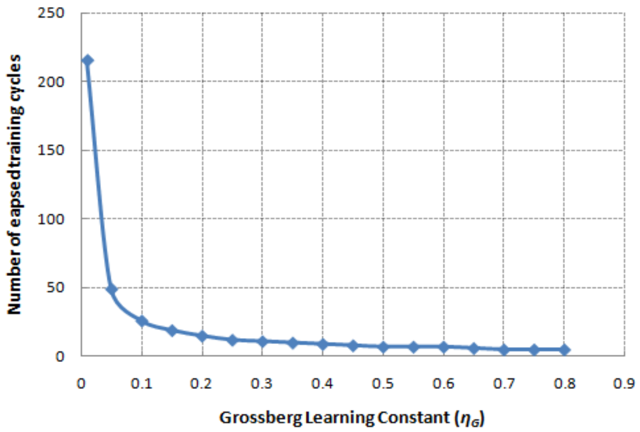
Network Topology (Number of Computing Units)			Kohonen Learning Constant (η_K)	Grossberg Learning Constant (η_G)	Error Function (E)	Number of Elapsed Training Cycle	Accuracy
Input Layer	Hidden Layer	Output Layer					
4	7	6	0.01	0.01	9.640942×10^{-5}	385	98%
			0.05		9.953296×10^{-5}	260	98%
			0.1		6.296407	903	79%
			0.15		9.865968×10^{-5}	232	98%
			0.2		9.699182×10^{-5}	231	99%
			0.25		9.855741×10^{-5}	223	99%
			0.3		9.901559×10^{-5}	215	100%
			0.35		3.305425	864	89%
			0.4		3.305425	904	87%
			0.45		3.305425	908	85%

TABLE III: RESULTS OF TUNING GROSSBERG LEARNING CONSTANT H_G KEEPING H_K FIXED

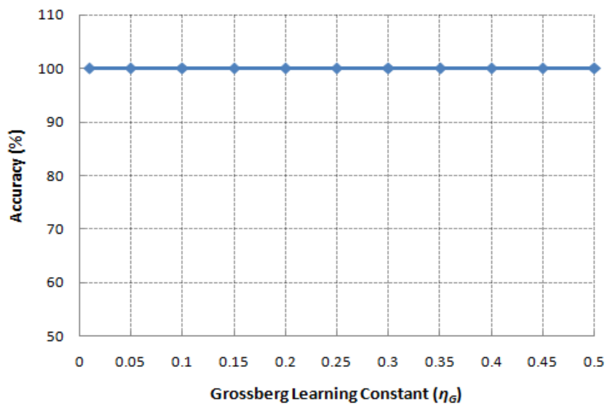
Network Size (No. of Computing Units)			Kohonen Learning Constant (η_K)	Grossberg Learning Constant (η_G)	Error Function (E)	Number of Elapsed Training Cycle	Accuracy
Input Layer	Hidden Layer	Output Layer					
4	7	6	0.3	0.01	9.901559×10^{-5}	215	100%
				0.05	9.743650×10^{-5}	49	100%
				0.1	9.074275×10^{-5}	26	100%
				0.15	5.553059×10^{-5}	19	100%
				0.2	4.864842×10^{-5}	15	100%
				0.25	8.012008×10^{-5}	12	100%
				0.3	3.174095×10^{-5}	11	100%
				0.35	1.884223×10^{-5}	10	100%
				0.4	1.760451×10^{-5}	9	100%
				0.45	2.739582×10^{-5}	8	100%
				0.5	7.599829×10^{-5}	7	100%
				0.55	1.676223×10^{-5}	7	100%
				0.6	3.111813×10^{-6}	7	100%
				0.65	3.095858×10^{-5}	6	100%
				0.7	7.209476×10^{-5}	5	100%
				0.75	1.627581×10^{-5}	5	100%
				0.8	2.666661×10^{-5}	5	100%



(a)



(b)



(c)

Figure 5. Results of tuning grossberg learning constant η_G keeping η_K fixed. (a) Error function (E). (b) Number of elapsed training cycle. (c) Accuracy.

TABLE IV. RESULTS OF THE COMPARISON OF OUR PREPROCESSED CPN MODEL AND OTHERS' MODELS

Reference	Type of Fabric	Number of Input Sites	Number of Classes	Sample Size (No. of Feature Vectors)	Performance Metrics			
					Training Time (Number of Elapsed Cycle)	Model Complexity		Accuracy
						Number of Computing Units	Connectivity	
Our work	Knitted fabric	4	6	100	5	4-7-6	Fully connected feedforward	100%
[4]	Knitted fabric	4	6	100	6	4-7-6	Fully connected feedforward	98.97%
[6]	Knitted fabric	4	6	100	191	4-12-6	Fully connected feedforward	100%
[7]	Knitted fabric	4	6	100	69937	4-15-6	Fully connected feedforward	100%
[8]	Knitted fabric	4	6	100	88811	4-12-6	Fully connected feedforward	100%
[9]	Knitted fabric	15	7	140	7350	15-8-7	Feedforward	78.4%
		NM	NM	8485	NM	NM	Feedforward	96.57%
[10]	Web textile fabric	13	8	400	NM	13-5-8	NM	94%
[12]	Web textile fabric	4	8	400	NM	4-5-8	NM	91%
[13]	NM	3	4	Over 200	NM	3-40-4-4	Fully connected feedforward	77%
[14]	NM	3	4	220	NM	3-44-4	Fully connected feedforward	76.5%
[15]	Knitted fabric	7	7	205	NM	7-7	NM	90.21%
		6	7	205	NM	6-7	NM	91.9%
[22]	Knitted fabric	4	6	100	50	4-26-6	Fully connected feedforward	91.75%

NM: Not Mentioned

Likewise we first train as well as test the CPN for η_G = 0.01 and η_K = 0.3. We raise the value of η_G slowly, and train as well as test the ANN for that value of η_G keeping the value of η_K unchanged. The results thus obtained are shown in Table III and Fig. 5. We observe that E is tolerable and accuracy is maximum for all η_G values tuned, i.e. $0.01 \leq \eta_G \leq 0.8$. The number of elapsed training cycle is minimum, i.e. 5 for $0.7 \leq \eta_G \leq 0.8$. 0.75 is chosen as the value of η_G since E is minimum for $0.7 \leq \eta_G \leq 0.8$.

Our aspiration was to find such values of Kohonen and Grossberg learning constants, for which minimum training time is elapsed and maximum classification accuracy is achieved. From this empirical quest, it appears that minimum training time is elapsed and maximum classification accuracy is achieved for η_K = 0.3 and η_G = 0.75.

VI. Results and Discussion

In order to assess merits of our PCPN model for classifying textile defects, we compare the results found and some recently reported relevant research results. Here it can be mentioned that there will be profound impacts on our attempt of comparative performance evaluation for the assumptions taken by researchers in collecting samples and reporting results of their research activities in processing those samples. The literature review discloses that most of research reports are limited to the demonstration of concepts of machine vision based approach to textile defect classification without the support of adequate numerical results and their comparison with similar works. A quantitative comparison between the various defect detection schemes is difficult as the performance of each of these schemes have been assessed/reported on the fabric test images with varying resolution, background texture and defects.

That means, the absence of use of common database of samples of textile defects makes it hard to have a fair comparison of merits of different algorithms.

Comparative performance evaluation based on realistic assumptions is not adequate although there have been some inspiring trends in textile defect inspection research for several years. We have tried to review numerical results related to textile defect classification to assess comparative merits of our work in spite of such limitations.

Table IV shows the comparison of our PCPN model and others' ANN models. It has been found in [31] that more than 95% accuracy appears to be industry benchmark. With respect to such observation, the accuracy of 100% obtained in our work appears to be good enough. Moreover, our model complexity (4, 7 and 6 computing units in the input, hidden and output layer respectively) and training time (5 cycles) have been intriguing enough. As we have mentioned earlier that it is not discerning enough to explicitly compare merits of our approach with other works due to the lack of uniformity in the image data set, performance evaluation and the nature of intended application. Therefore, it does not seem to be unfair to claim that our implemented PCPN model has enough potential to classify textile defects with very good accuracy.

VII. Conclusion

We have presented a modified, i.e. preprocessed CPN classifier in the domain of fabric defect classification in this work. Although the literature review reveals that the CPN classifier has been found suitable for automated fabric defect classification, we have found increase in performances by introducing PCPN classifier. It's believed that this PCPN classifier will be laying the basis to guide application engineers to decide about which classifier to apply for defect classification within short time. Finally, we have compared the performances of our PCPN model with that of the classification models described in different articles. In comparison to classification performances of reported research findings, the 100% accuracy, simple model complexity (4, 7 and 6 computing units in the input, hidden and output layer respectively) and short training time (5 cycles) in classifying commonly occurring fabric defects in Bangladesh appears to be excellent.

Enough investigation has not been performed on data pattern. There remains a research challenge whether this preprocessed approach is applicable to data of all patterns. Moreover, the findings of our work are not comprehensive enough to make conclusive comments about the merits of PCPN classifier, because sample size was not large enough.

Lighting was not good enough to acquire very high quality images. Further work remains not only to successfully classify commonly occurring fabric defects in Bangladesh for a sample of a very large number of high-quality images but also to deal with classification problems in different domains.

References

[1] M. F. Zafar, D. Mohamad, and R. M. Othman, "On-Line handwritten character recognition: An implementation of counterpropagation neural net," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 10, December 2005.

[2] M. F. Zafar and D. Mohamad, "Counter propagation neural networks for trademark recognition," in *Proc. International Symposium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia, 13-I6 August, 2001.

[3] K. Sridharan and J. Abraham, "A counter propagation neural network for face detection and recognition," in *Proc. National Conference on Signal Processing Intelligent Systems and Networking*, Bangalore, India, December 2003, pp. 65-68.

[4] M. T. Habib and M. Rokonuzzaman, "An empirical method for optimization of counterpropagation neural network classifier design for fabric defect inspection," *International Journal of Intelligent Systems and Applications*, vol. 6, no. 9, pp. 30–39, August 2014.

[5] Trade Information. [Online]. Available: http://www.bgmea.com.bd/home/pages/tradeinformation#.VJVlQkY4Bg

[6] M. T. Habib and M. Rokonuzzaman, "A set of geometric features for neural network based textile defect classification," *International Scholarly Research Network Artificial Intelligence*, 2012.

[7] M. T. Habib and M. Rokonuzzaman, "An empirical approach to optimize design of backpropagation neural network classifier for textile defect inspection," *SCIENCEDOMAIN British Journal of Mathematics & Computer Science*, vol. 3, no. 4, pp. 617–634, October-December 2013.

[8] M. T. Habib and M. Rokonuzzaman, "Distinguishing feature selection for fabric defect classification using neural network," *Academy Publisher Journal of Multimedia*, vol. 6, no. 5, pp. 416–24, 2011.

[9] R. G. Saeidi, M. Latifi, S. S. Najar, A. G. Saeidi, "Computer vision-aided fabric inspection system for on-circular knitting machine," *Textile Research Journal*, vol. 75, no. 6, pp. 492-497, 2005.

[10] Y. A. Karayiannis, R. Stojanovic, P. Mitropoulos, C. Koulamas, *et al.*, "Defect detection and classification on web textile fabric using multiresolution decomposition and neural networks," in *Proc. 6th IEEE International Conference on Electronics, Circuits and Systems*, Pafos, Cyprus, September 1999.

[11] C. F. J. Kuo and C. J. Lee, "A back-propagation neural network for recognizing fabric defects," *Textile Research Journal*, vol. 73, no. 2, pp. 147-151, 2003.

[12] P. Mitropoulos, C. Koulamas, R. Stojanovic, S. Koubias, G. Papadopoulos, and G. Karayiannis, "Real-Time vision system for defect detection and neural classification of web textile fabric," in *Proc. SPIE*, San Jose, California, January 1999, pp. 59-69.

[13] M. A. Islam, S. Akhter, and T. E. Mursalin, "Automated fabric defect recognition system using computer vision and artificial neural networks," *Proceedings World Academy of Science, Engineering and Technology*, vol. 13, pp. 1-7, 2006.

[14] M. A. Islam, S. Akhter, T. E. Mursalin, and M. A. Amin, "A suitable neural network to detect fabric defects," in *Proc. Neural Information Processing*, 2006, pp. 430-438.

[15] E. Shady, Y. Gowayed, M. Abouiiana, S. Youssef, and C. Pastore, "Detection and classification of defects in knitted fabric structures," *Textile Research Journal*, vol. 76, no. 4, pp. 295-300, 2006.

[16] K. L. Mak, P. Peng, and H. Y. K. Lau, "A real-time computer vision system for detecting defects in textile fabrics," in *Proc. IEEE International Conference on Industrial Technology*, Hong Kong, China, December 2005.

[17] A. Baykut, A. Atalay, A. Erçil, and M. Güler, "Real-Time defect inspection of textured surfaces," *Real-Time Imaging*, vol. 6, no. 1, pp. 17–27, 2000.

[18] A. Kumar, "Neural network based detection of local fabric defects," *Pattern Recognition*, vol. 36, pp. 1645-1659, 2003.

[19] F. S. Cohen and Z. Fan, "Rotation and scale invariant texture classification," in *Proc. IEEE Conf. Robot. Autom*., vol. 3, pp. 1394–1399, April 1988.

[20] D. A. Karras, S. A. Karkanis, and B. G. Mertzios, "Supervised and unsupervised neural network methods applied to textile quality control based on improved wavelet feature extraction techniques," *International Journal on Computer Mathematics*, vol. 67, pp. 169-181, 1998.

[21] R. Stojanovic, P. Mitropulos, C. Koulamas, Y. A. Karayiannis, S Koubias, and G. Papadopoulos, "Real-time vision based system for textile fabric inspection," Real-Time Imaging, vol. 7, no. 6, pp. 507–518, 2001.

[22] M. T. Habib, R. H. Faisal, and M. Rokonuzzaman, "Feasibility of genetic algorithm for textile defect classification using neural network," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 4, pp. 131–150, July 2012.

[23] A. Abouelela, H. M. Abbas, H. Eldeeb, A. A. Wahdan, S. M. Nassar, "Automated vision system for localizing structural defects in textile fabrics," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1435-1443, 2005.

[24] R. Furferi, L. Governi, and Y. Volpe, "Neural network based classification of car seat fabrics," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 5, no. 3, pp. 696–703, 2015.

[25] R. Furferi and L. Governi, "The recycling of wool clothes: an artificial neural network colour classification tool," *International Journal of Advanced Manufacturing Technology*, vol. 37, no. 7-8, pp. 722–731, 2008.

[26] R. Hecht-Nielsen, "Counterpropagation networks," *Applied Optics*, vol. 26, no. 23, pp. 4979-4983, 1987.

[27] D. Anderson and G. McNeill, "Artificial neural networks technology," Contract Report, for Rome Laboratory, contract no. F30602-89-C-0082, August 1992.

[28] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed., Academic Press, 2008.

[29] D. Phillips, *Image Processing in C*, 2nd ed. Kansas: R & D Publications, 2000.

[30] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, USA: Pearson Education, Inc., 2006.

[31] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 348-363, 2008.

Md. Mokhlesur Rahman was born at Satkhira in Bangladesh. He received his B.Sc degree in Computer Science & Engineering from Mawlana Bhashani Science and Technology University, Bangladesh in 2010. His research interest includes in Artificial Intelligence.

He worked as a Senior Web Developer at Squadro Solutions, Dhaka, Bangladesh in 2010. In the next year, he joined to International Acumen Limited as an Assistant Engineer, Dhaka, Bangladesh. At the end of 2012, he joined as a LECTURER at Prime University, Mirpur-1, Dhaka, Bangladesh.

Md. Tarek Habib got his B.Sc. degree in Computer Science from BRAC University in 2006. Then he obtained M.S. degree in Computer Science and Engineering (Major in Intelligent Systems Engineering) from North South University in 2009. Now he is an Assistant Professor at the Department of Computer Science and Engineering in Green University of Bangladesh. He has had a number of publications in international and national journals and conference proceedings. His research interest is in Artificial Intelligence, especially Artificial Neural Networks, Pattern Recognition, Computer Vision and Natural Language Processing.