PERGAMON

Pattern Recognition 36 (2003) 1645–1659

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Neural network based detection of local textile defects

Ajay Kumar*

Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Received 5 April 2002; accepted 28 October 2002

Abstract

A new approach for the segmentation of local textile defects using feed-forward neural network is presented. Every fabric defect alters the gray-level arrangement of neighboring pixels, and this change is used to segment the defects. The feature vector for every pixel is extracted from the gray-level arrangement of its neighboring pixels. Principal component analysis using singular value decomposition is used to reduce the dimension of feature vectors. Experimental results using this approach illustrate a high degree of robustness for the detection of a variety of fabric defects. The acceptance of a visual inspection system depends on economical aspects as well. Therefore, a new low-cost solution for the fast web inspection using linear neural network is also presented. The experimental results obtained from the real fabric defects, for the two approaches proposed in this paper, have confirmed their usefulness.

Keywords: Defect detection; Machine vision; Automated visual inspection; Quality assurance; Neural networks

1. Introduction

Automated visual inspection of industrial goods for quality control plays an ever-increasing role in production process as the global market pressures put higher and higher demand on quality. In most cases, the quality inspection through visual inspection is still carried out by humans. However, the reliability of manual inspection is limited by ensuing fatigue and inattentiveness. For example in textile industry, the most highly trained inspectors can only detect about 70% of the defects [1]. Therefore, the automation of visual inspection process is required to maintain high quality of products at high-speed production.

Some of the most challenging visual inspection problems deal with the textured materials. Three common criterion used to measure the quality index of textured materials are related to material isotropy, homogeneity and texture coarseness [2]. While there is a remarkable similarity in the overall automation requirements for the textured materials, the cost effective solutions are problem specific and require extensive research and development efforts. Quality assurance in production lines for textured materials such as wood [3], steel-roll [4], paper [5], carpet [6], textile [1,7–21], etc., have been studied by various researchers. The detection of local fabric defects is one of the most intriguing problems in visual inspection, and has received much of the attention over the years [7–21]. This paper focuses on this problem and investigates some new techniques to address this problem.

1.1. Prior work

Fabric defect detection using digital inspection images has received considerable attention during the past decade and numerous approaches have been proposed in the literature [7–21]. At microscopic level, the inspection problems encountered in digital images become texture analysis problems. Therefore texture features based on statistical, geometrical, structural, model based, or signal-processing approaches are the potential source of investigation [22]. In the approach by Cohen et al. [7] Gauss Markov Random Field (GMRF) model has been used for the characterization of fabric textures. The web inspection problem is

* Tel.: +852-23-58-8384; fax: +852-23-58-1477.
E-mail address: ajaykr@cs.ust.hk (A. Kumar).

0031-3203/03/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0031-3203(03)00005-0

treated as hypothesis-testing problem on the statistics derived from this model. Campbell et al. [8] have performed fabric defect detection on denim fabrics using model based clustering. Ade [9] obtained the principal components from the variance–covariance matrix of fabric texture and used them for defect detection. Characterization of fabric defects based gray-level co-occurrence matrix [10], mean and standard deviation of sub blocks [11], Karhunen–Loéve (KL) transform [12], optimal filters [13,23] and autocorrelation of non-overlapping sub-images [14] have been reported in the prior work. Chetverikov [24] has introduced the quantitative definition of maximum regularity and shown it to be consistent with human judgment on regularity of textures. Recently, the maximum regularity and orientation coherence have been used in Ref. [25] for the detection of local textured defects.

Every fabric defect causes some change in 2-D symmetry of fabric texture, which results in various changes in its optical Fourier transform (OFT). Therefore, a combination of OFT and neural networks has been used to detect and identify the fabric defects [15]. Fourier transform based techniques [16] are successful for defects that cause a global change of fabric texture, but are not suitable for local defects that usually occur in small area of inspection images. Detection of local fabric defects requires multi-resolution decomposition of inspection image across several scales. Therefore, texture features based on multi-scale wavelet decomposition have been used [1]. Jasper et al. [17] use texture adapted wavelet bases, which offer high sensitivity to the abrupt change in fabric texture caused by defects. Escofet et al. [18] have detailed fabric defect detection using a bank of multi-scale Gabor filters. Defect detection using only real Gabor functions has been detailed [19]. Defect detection techniques using 2-D image analysis are computationally intensive and time consuming. Therefore, Kim et al. [20] have used 1-D projection signals generated from the 2-D images. Fabric defects have been detected with the multi-scale wavelet analysis on such 1-D signals. However, 1-D signals used in Ref. [20] does not preserve adjancy of features, except in the two directions of scanning. Therefore, 1-D signals generated from the inspection images using factual scanning has been used for fabric defect detection and identification [21].

1.2. Present work

In this paper, a new approach for fabric defect detection using Feed-Forward Neural network (FFN) is investigated. Neural networks are among the best classifiers used for fault detection due to their non-parametric nature and ability to describe complex decision regions. Although there are several examples on the applications of neural networks to the problem of defect detection [10,14], no work has been reported on the segmentation of defects. The application of neural network for fabric defect segmentation detailed in this paper has been guided by two factors: computational simplicity and robustness. The computational simplicity is critical for success of any approach for real time fabric inspection. The dimension and orientation of fabric defects on the textile web varies randomly. Therefore, the performance of any defect detection scheme depends on its robustness for the detection of majority of defects, and this has been evaluated in this work.

The main contributions [13] of this paper are summarized below:

(i) A new approach for the fabric defect segmentation using FFN has been investigated. Excellent defect segmentation results on real fabric samples using this approach are demonstrated.
(ii) A low-cost solution for fabric defect detection using linear neural network is proposed. Experimental results on the low-resolution images obtained from the real fabric samples confirm its usefulness.

The organization of this paper is as follows: In Section 2, the approach for fabric defect segmentation using FFN is described. The details on this approach appear in several subsections following Section 2. These subsections include a review on principal component analysis, singular value decomposition and its application in principal component analysis. This is followed by a brief summary of FFN in Section 2.3. In Section 2.4 details of the experiments conducted on fabric samples using FFN are described. The experimental results and the discussion on these results appear in Section 2.5. In Section 3, the low-cost fabric defect detection approach using linear neural networks is introduced. This approach is detailed in four subsections. Section 3.1 presents a brief review on linear neural network, followed by methodology of this approach in Section 3.2. The experimental results of this approach are reported in Section 3.3, which is followed by discussion in Section 3.4. Finally, conclusions from this work are summarized in Section 4.

2. Fabric defect segmentation using FFN

Inspection of textile webs using the segmentation of defects in the inspection images is most common and is described in Refs. [1,7–9,11,12,17–19]. Segmentation of defects provides accurate localization of size and location of defects. The block diagram of this approach using FFN is shown in Fig. 1. In this approach, FFN is used to classify feature vectors of pixels obtained from the real inspection images. The real data often suffers from a number of deficiencies such as missing input value or incorrect target values [26]. Therefore, some form of pre-processing of feature vectors before their application to FFN is needed. In this work, the pre-processing using (i) normalization of feature vectors followed by their (ii) principal component analysis

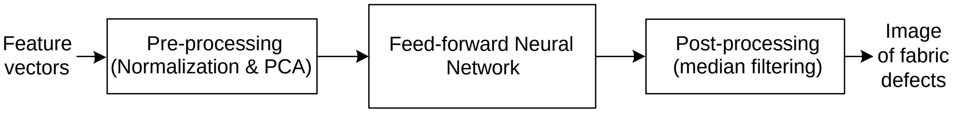
Feature vectors
Pre-processing (Normalization & PCA)
Feed-forward Neural Network
Post-processing (median filtering)
Image of fabric defects

Fig. 1. Block diagram of Fabric defect segmentation using FFN.

is used. The output from FFN is post-processed to generate the required output value. This approach is detailed in the following subsections.

2.1. Extraction of feature vectors

A key issue in many neural network applications is to determine which of the available input features should be used for modeling. Texture is regarded as neighborhood property of an image that contains variations of intensities that exhibit a uniform or random pattern. Defect detection in a given image requires identification of regions with *uniform* texture. Harlick et al. [27] have proposed a number of statistical and structural features to describe a texture, but the problem of efficient representation and description is not yet solved. An exhaustive evaluation of potential feature sets for the characterization of texture is difficult. Therefore, feature selection based on simple fast evaluating models and heuristic selection strategies has been used in this work.

Every pixel from the acquired image is characterized by a feature vector. A simplified characterization of texture based on relations between the pixels in restricted neighborhood is used. The gray-level values of neighboring pixels form the feature vector for every pixel. The feature vector does not pretend to provide an exhaustive characterization of texture, but they do implicitly capture certain local textural properties such as coarseness, directionality, regularity, etc. The three neighborhood configurations, known as mask or macro-window, as shown in Fig. 2, were considered in this study. As shown in Fig. 2(a), the central pixel (shown black) is characterized by shaded neighboring pixels in a 7 × 7 macro-window. The size of macro-window should be large enough to capture the characteristic gray-level variations of the texture, and small enough to base the decision about the pixels texture (defect-free or from defect) on its neighborhood only [28]. Large macro-windows increase the size of feature vector for every pixel, and are therefore computationally undesirable for the real time inspection. The macro-window configuration shown in Fig. 2(c) has been used by Mao and Jain [29] in their autoregressive texture model, and was found to be the most suitable among the three as detailed in Section 2.5. Thus every pixel in an acquired *sub-image* [1] is characterized by a feature vector consisting of gray-level values from the 25 pixels in a 7 × 7 macro-window (Fig. 2.c). These feature vectors form the input to the neural network.

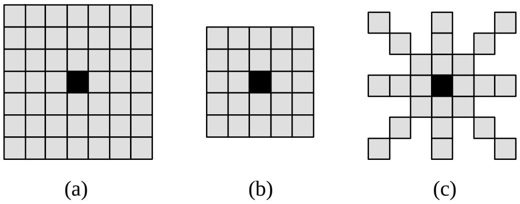

Fig. 2. Extraction of feature vector for every pixel: 49, 25, and 25 features of a pixel from its neighboring pixels using 7 × 7, 5 × 5, and 7 × 7 macro-window in (a), (b) and (c), respectively.

The large dimension of feature vectors generates computational and over-fitting problems. Now we are faced with two conflicting goals. On one hand, it is desirable to simplify the problem by reducing the dimension of feature vectors. On the other hand, the original information content has to be preserved as much as possible. The principal component analysis offers a convenient way to control the tradeoff between simplifying the problem and loosing the information, and is reviewed in the following section.

2.2. Principal component analysis

Principal component analysis or Karhunen–Loève (KL) transform is a de-correlation based method that finds a linear transformation $\mathbf{W}$ on a multivariate data $\mathbf{X}$,

$$\mathbf{X}' = \mathbf{W}\mathbf{X}. \tag{1}$$

The transformation $\mathbf{W}$ is such that (a) the output vectors of $\mathbf{X}'$ are uncorrelated, (b) the basis vectors of $\mathbf{W}$ are orthogonal to each other and, (c) the eigenvalues of $\mathbf{W}$ are ordered. The transformation $\mathbf{W}$ is computed using eigenvalue decomposition (ED), which is a common statistical tool for the analysis of multivariate data. However, the eigenvalue decomposition is applicable only on symmetric square matrices, but the singular value decomposition (SVD) can be applied to any matrix, regardless of its size, symmetric or non-symmetric, rank deficient, etc. The SVD is detailed in next subsection.

2.2.1. Singular value decomposition

Let $\mathbf{X}$ be an arbitrary real $p \times q$ matrix, i.e. $\mathbf{X} \in \mathscr{R}^{p\times q}$. The singular value decomposition of matrix $\mathbf{X}$ is given by [30–32],

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}, \tag{2}$$

[1] The portion of inspection image for which the feature vectors can be constructed, i.e. by ignoring few (three) boundary pixels.

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrix with $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p] \in \mathscr{R}^{p\times p}$, and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_q] \in \mathscr{R}^{q\times q}$. The $\mathbf{\Sigma}$ is an $p \times q$ diagonal matrix with non-negative, real elements σ_j in the descending order,

$$\mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k), \tag{3}$$

where $k = \min(p, q)$ and, $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_k \geqslant 0$. The matrix $\mathbf{\Sigma}$ is padded with zeros, such that the dimension of the products of matrices in Eq. (2) is same as the dimension of matrix $\mathbf{X}$, i.e.

$$\begin{bmatrix} \mathbf{\Sigma} \\ 0 \end{bmatrix} \quad \text{if } p > q \quad \text{or} \quad [\mathbf{\Sigma}\ 0] \quad \text{if } p < q. \tag{4}$$

The values σ_j in Eq. (3) are referred to as singular values of $\mathbf{X}$. The columns $\mathbf{u}_j$ of $\mathbf{U}$ are called as left singular vectors of $\mathbf{X}$ and, the columns $\mathbf{v}_j$ of $\mathbf{V}$ are called as right singular vectors of $\mathbf{X}$. Using Eq. (2),

$$\mathbf{XX}^{\mathrm{T}} = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^{\mathrm{T}}. \tag{5}$$

It can be observed from Eq. (5) that the eigenvectors of matrix $\mathbf{XX}^{\mathrm{T}}$ are the left singular vectors $\mathbf{u}_j$ of $\mathbf{X}$ and, the nonzero eigenvalues of matrix $\mathbf{XX}^{\mathrm{T}}$ are the squared singular values of $\mathbf{X}$.

Multivariate populations and feature vectors are often characterized by covariance matrix. The $p \times q$ matrix $\mathbf{X}$ can be formed from the p observations on each of the q feature vectors, i.e. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_q]$. An estimate $\hat{\mathbf{C}}$ of covariance matrix corresponding to these $\mathbf{x}_q$ feature vectors can be obtained as follows [31]:

$$\hat{\mathbf{C}} = \frac{1}{q}[\mathbf{XX}^{\mathrm{T}}]. \tag{6}$$

Using Eqs. (5) and (6) it can be stated that the left singular vectors $\mathbf{u}_j$ of $\mathbf{X}$ are the eigenvectors of covariance matrix $\hat{\mathbf{C}}$, and the singular values of $\mathbf{X}$ are the positive square roots of the eigenvalues of covariance matrix $\hat{\mathbf{C}}$. Thus the eigenvalues and the eigenvectors of a covariance matrix $\hat{\mathbf{C}}$ can be obtained from the SVD of $\mathbf{X}$.

For a square symmetric matrix $\mathbf{A}$, using Eq. (2),

$$\mathbf{A} = \mathbf{U\Sigma V}^{\mathrm{T}} \quad \text{or} \quad \mathbf{A}^{\mathrm{T}} = \mathbf{V\Sigma U}^{\mathrm{T}}. \tag{7}$$

Since $\mathbf{A} = \mathbf{A}^{\mathrm{T}}$ and therefore $\mathbf{U} = \mathbf{V}$. The SVD and ED of a square symmetric matrix, such as $\hat{\mathbf{C}}$, are equivalent [31]. The SVD is computationally more reliable than the ED [32]. Therefore, instead of computing ED of $\hat{\mathbf{C}}$ (or $\mathbf{XX}^{\mathrm{T}}$), SVD of $\mathbf{X}$ is used to compute the principal components as detailed in next subsection.

2.2.2. *Principal components using SVD*

Geometrically the eigenvectors or the singular vectors can be considered as the axes of new coordinate system. As discussed in last subsection, the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{XX}^{\mathrm{T}}$ corresponding to non-zero eigenvalues [31]. These eigenvectors are known as principal components, and are of potential interest since they can be used to compute the components of transformed data $\mathbf{X}'$ (1) by projecting the original multivariate data onto them. The orthogonal projection of input vectors from $\mathbf{X}$ onto the columns of $\mathbf{U}$ is obtained by following matrix transformation:

$$\mathbf{X}' = \mathbf{U}^{\mathrm{T}}\mathbf{X}. \tag{8}$$

The columns of transformed matrix $\mathbf{X}'$ are the vectors whose elements are uncorrelated and ordered by decreasing variance. The correlation matrix of transformed data $\mathbf{X}'$ is a diagonal matrix $\mathbf{\Sigma}$ whose elements are comprised of ordered singular values. Comparing Eqs. (1) with (8) yields, $\mathbf{W} = \mathbf{U}^{\mathrm{T}}$. The transformed data points are the linear combinations of original data values weighted by the eigenvectors of $\mathbf{XX}^{\mathrm{T}}$. Each column of eigenvector matrix $\mathbf{U}$ represents the weight applied to the input data points (columns of $\mathbf{X}$) to build each of the respective transformed data point. The percentage of total variance in each of the principal components can be computed from the elements of $\mathbf{\Sigma}$ as follows:

$$\%\ \text{Variance} = \frac{\sigma_j}{\sum_{j=1}^{k} \sigma_j}. \tag{9}$$

Since the diagonal elements of $\mathbf{\Sigma}$ are arranged in descending order, the first principal component has the largest percentage of total variance. Each of the subsequent components contains the maximum variance for any axes orthogonal to the previous component. Those principal components, which contribute very small amount of variance, can be neglected. This is achieved by neglecting those columns of $\mathbf{U}$ in equation (8), which contribute to small variance σ_j. Hence the transformation with reduced $\mathbf{U}$ (columns of $\mathbf{U}$) results in transformed data $\mathbf{X}'$ whose dimension is smaller than the dimension of $\mathbf{X}$.

2.3. *Feed-forward neural network*

Execution speed of multi-layer feed-forward neural network is among the fastest of all models currently in use. Therefore, this network may be the only practical choice for real-time inspection of defects in textured materials. A feed-forward neural network with N_l neurons in the lth $(l = 1, \ldots, M)$ layer is based on the following architecture:

$$\varphi_j^l = \sum_{i=1}^{N_{l-1}} w_{ij}^{l-1,l}\, y_i^{l-1}, \qquad y_j^l = g(\varphi_j^l), \tag{10}$$

where the sum of weighted inputs for jth $(j = 1, \ldots, N_l)$ neuron in the lth layer is represented by φ_j^l. The weights from the ith neuron at $(l-1)$th layer to the jth neuron in the lth layer are denoted by $w_{ij}^{l-1,l}$ and y_j^l is the output for jth neuron in the lth layer. The values -1 and 1, corresponding to 'defect' and 'defect-free' responses, were given to the network during training as the correct output responses expected classification (pixel gray level values) during the training. For a two-layer network used in this work, hyperbolic tangent sigmoid activation function was chosen since its output $(-1$ to $1)$ is perfect for learning target output values $(-1$ to $1)$.

$$g(\varphi_j^l) = \tanh(\varphi_j^l) \quad \text{for } l = 1, 2. \tag{11}$$

The back-propagation training algorithm is used for minimizing training function E defined by

$$E = \frac{1}{KN_M} \sum_{k=1}^{K} \sum_{j=1}^{N_M} (y_{j,k}^{M} - o_{j,k})^2 \qquad (12)$$

where k is an index for input–output pair and $(y_{j,k}^{M} - o_{j,k})^2$ is the squared difference between the actual output value at the jth output layer neuron for pair k and the target output value. The connection weights $w_{ij}^{l-1,l}$ are updated after presentation of every feature vector using a constant learning rate. The weights are updated using Levenberg–Marquardt [33] algorithm for faster convergence rate. The details of this algorithm can be found in Refs. [33,34].

2.4. Experiments

Twill and plain weave fabric samples of most commonly occurring fabric defects were gathered from the loom. All images were acquired under backlighting and cover 1.28 × 1.28 in^2 area of fabric sample. The acquired images are digitized in 256 × 256 pixels with 8-bit resolution (256 gray levels). A pair of these histogram-equalized images, one with fabric defect (D) and other without any defect (R), is used for training FFN. The feature vectors for training are extracted from (i) a small image pitch of typically 40 × 40 pixels in the region of image having defect in D and, (ii) equal image pitch pixels from R. Every feature vector is normalized to have zero mean and unity variance. Each of these normalized feature vectors are subjected to PCA, i.e. using Eq. (8) the transformed feature vectors (columns of $\mathbf{X}'$) are obtained. The dimensions of feature vectors are substantially reduced with the pre-processing, i.e. normalization and PCA, and the resulting training vectors are presented to FFN (Fig. 1). The FFN with 5 input nodes and 1 output node (5/1) was employed in supervised training mode, i.e. target value of 1 was used for the features from a defect free pixel and -1 for pixels with defect. The network is trained for the maximum of 1000 steps with the learning rate of 0.01 and the training is stopped if the maximum performance gradient of $1\mathrm{e}-10$ is reached. Due to a number of pixels covering the full scope of defect and defect-free region, the selection of appropriate training set is important. The objective of training paradigm is to obtain satisfactory performance of FFN with as small-as-possible training set. There is no guarantee that the achieved training error is global and therefore avoidance of local minima is important. In this work, the FFN was trained 20 times with the same parameters and the result with the smallest of training errors of all the results are reported. Once the network is trained, the whole of the 256 × 256 pixels image with fabric defects is used to test the performance of the network. The post-processing operation consisted of a 9 × 9 median filtering to reduce speckle-like noise [28] in the output image.

2.5. Results and discussion

A twill weave fabric sample with defect *mispick* as shown in Fig. 3(a) was used to compute principal components using SVD. The 20 × 80 pixel image pitch from each of the defect-free and defect image was used to obtain feature vectors using three separate macro-windows shown in Fig. 2. From this image (pitch) feature vector matrix of dimension 49 × 2072, 25 × 2432, and 25 × 2072 was generated by using macro-window configuration shown in Fig. 2(a), (b) and (c), respectively. Each column of these feature vector matrices represent a feature vector corresponding to a pixel. Using the SVD, this feature vector matrix (say $\mathbf{X}$) is decomposed into three separate matrices, i.e. $\mathbf{X} = \mathbf{U\Sigma V}^{\mathrm{T}}$. As detailed in Section 2.2.2, for each of the three macro-windows, we can now calculate the amount of variability in each of the principal components using Eq. (9). The results of SVD on each of feature vector matrix corresponding to three macro-windows in Fig. 2 are summarized in Table 1.

From Table 1, it can be seen that for the feature vector matrix obtained by using macro-window of Fig. 2(a), the first 13 principal components account for 99% of the total variability in the transformed feature vectors (matrix $\mathbf{X}'$ in Eq. (8)). Thus the remaining 36 components account for only 1% of the total variance in the transformed feature vectors (columns of $\mathbf{X}'$). A similar interpretation of results in other columns of this table allows us to make tradeoff in the dimension of (transformed) feature vector matrix with the amount of variability/information in the (transformed) feature vectors. A similar analysis for all three macro-windows in Fig. 2, when image under inspection are not histogram equalized is also presented in Table 1. It can be observed from this table that for the same amount of total variance in the principal components, fewer components are required when the images are histogram equalized. Therefore, histogram equalization of images before the inspection (training and testing) is justified as it will increase the speed of inspection. As a compromise between the two contradictory goals of possessing as much variance and reducing dimension of feature vectors, macro-window shown in Fig. 1(c) is chosen, with the principal components accounting for 98% of total variance, for the further results reported in this paper.

Thus for the *mispick* shown in Fig. 3(a), the dimension of 25 × 2072 feature vector matrix is reduced to 9 × 2072 and is used for training FFN. For the fixed dimension of feature vectors, FFN has two main parameters that influence its performance, speed and the accuracy of learning. These are number of layers and number of neurons used in each layer. Based on heuristics, feature vectors from the *mispick* were evaluated on three different network configurations: (a) two layer 9/1, (b) two layer 5/1, and (c) three layer 9/2/1. The convergence of training error for these three topologies is displayed in Fig. 4. For the 5/1 FFN, the training stops after 120 steps as the maximum performance gradient is reached with the increase in number of neurons from five to nine (equal to number of inputs), no significant improvement in

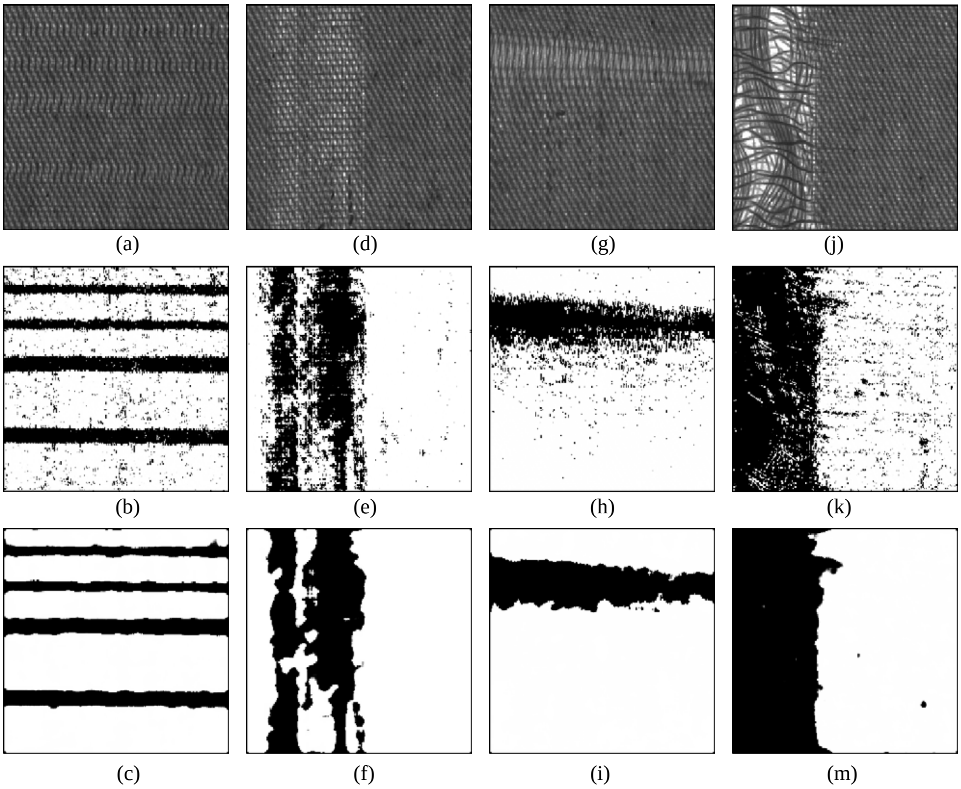

Fig. 3. Twill weave fabric samples with defect *mispick*, *netting-multiplies* (1), *thin bar*, and *netting-multiplies* (2), in (a), (d), (g) and (j) respectively; their corresponding segmented defects from 5/1 FFN outputs in (b), (e), (h) and (k), respectively; the corresponding segmented defects after postprocessing the FFN output images in (c), (f), (i) and (m), respectively.

Table 1
Principal component analysis of feature vector matrix for *mispick*

PVN	Case a: 49 × 2072		Case b: 25 × 2432		Case c: 25 × 2072	
	With HistEq	Without HistEq	With HistEq	Without HistEq	With HistEq	Without HistEq
1%	13 × 2072	16 × 2072	10 × 2432	11 × 2432	13 × 2072	16 × 2072
2%	9 × 2072	10 × 2072	8 × 2432	9 × 2432	9 × 2072	10 × 2072
3%	7 × 2072	9 × 2072	6 × 2432	7 × 2432	8 × 2072	8 × 2072

PVN: Percentage of total variance neglected. HistEq: Histogram Equalization.
Case a: 49 features from 7 × 7 macro-window in Fig. 2(a).
Case b: 25 features from 5 × 5 macro-window in Fig. 2(b).
Case c: 25 features from 7 × 7 macro-window in Fig. 2(c).

performance is observed. However, a three-layer 9/2/1 topology having two neurons in the second hidden layer achieved much smaller training error (Fig. 4), but this topology will require more computations and is therefore undesirable for real time inspection. Thus the two-layer network configuration with 5/1 has been used for further experiments with various fabric samples.

The segmentation of fabric defect *mispick* with trained 5/1 FFN is shown in Fig. 3(b). This image after post-processing (median filtering) is shown in Fig. 3(c). The segmentation results from other experiments on twill and plain weave fabric samples are shown in Figs. 3 and 5. The summary of training parameters for the FFN used in the experimental results of Figs. 3 and 5 appears in Table 2.

The FFN trained to produce the results shown in Fig. 3(b) was used to test the robustness of this segmentation scheme. This *mispick* trained FFN was tested with other (but twill weave, same texture) fabric samples having different defects. The test yielded excellent results as the *mispick* trained FFN could segment several other class of fabric defects such as

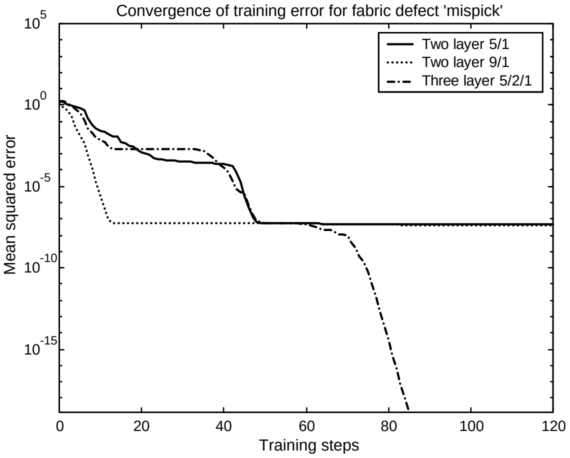

Fig. 4. Convergence of training error fro fabric defect *mispick*.

slack-end, *netting-multiplies*, *thin-bar*, *oil-stain*, etc. Fig. 6 shows segmentation of some of the different fabric defects with the mispick trained FFN. Comparing the segmentation of defect samples in Fig. 3 and their corresponding segmentation (with *mispick* trained FFN) in Fig. 6, it can be observed that the results in Fig. 6 (before post-processing) contain more noise. However, the segmentation is satisfactory after median filtering, although the segmented results are marginally poor as compared to those in Fig. 3. A similar set of experiments with the *mispick* trained (also with the *thin-bar* or *netting-multiplies* trained) FFN for the detection/segmentation of other fabric defects, which are not reproduced in this paper, have shown successful results. Since almost all the textile defects are characterized by significant change in gray-level arrangement of neighboring pixels, we can conclude that these defects can be detected successfully. However, the degree of segmentation (or the amount of noise) of defects varies from defect to defect. Three possible reasons can be attributed to this variation. Firstly, the texture of fabric under inspection is not exactly as same as that used for the training FFN. This can be due to (i) elasticity of yarns and/or, (ii) random variations in the alignment of fabric with respect to camera. Secondly, the impurities present in the yarns of the fabric also contribute to this noise. Thirdly, the wrinkles (crease) present in the fabric under inspection can alter gray-level arrangement of neighboring pixels, and hence generate false alarm or noise.

2.5.1. Real-time implementation

Can the FFN be implemented in real-time? Let us consider the 5/1 FFN with 25 inputs (ignoring PCA) used

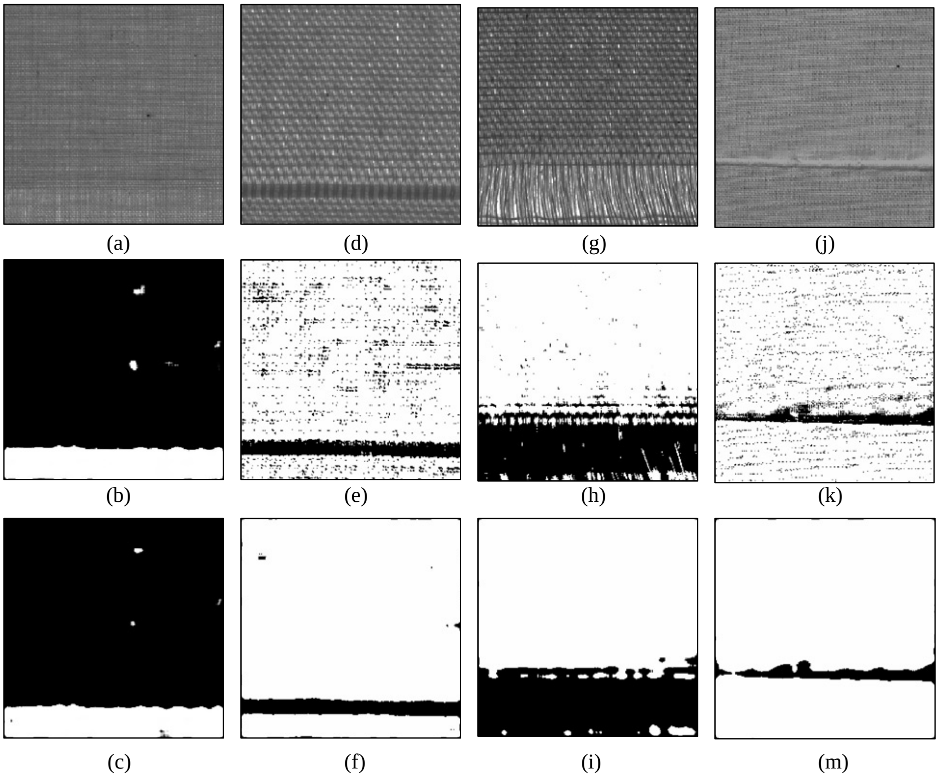

Fig. 5. Plain weave fabric samples with defect *double weft*, *thin bar*, *broken ends*, *sleck-pick* in (a), (d), (g) and (j), respectively; their corresponding segmented defects from 5/1 FFN outputs in (b), (e), (h) and (k), respectively; the corresponding segmented defects after post-processing the FFN output images in (c), (f), (i) and (m), respectively.

Table 2
Summary of some fabric defects and their training parameters used in the experiment

Figure no.	3(a)	3(d)	3(g)	3(j)	5(a)	5(d)	5(g)	5(j)
Name of fabric defect	Mispick	Netting-multiplies (1)	Thin-bar	Netting-multiplies (2)	Double-weft	Thin-bar	Broken-ends	Slack-pick
Weaving style	Twill	Twill	Twill	Twill	Plain	Plain	Plain	Plain
Yarn density Warp × weft (per inch)	118 × 60	118 × 60	118 × 60	118 × 60	136 × 72	84 × 28	84 × 28	136 × 72
Neural Network configuration	5/1	5/1	5/1	5/1	5/1	5/1	5/1	5/1
Size of image pitch for training samples	20 × 80	35 × 35	40 × 40	40 × 40	20 × 60	20 × 60	20 × 60	15 × 40
Size of feature vector matrix (**X**)	25 × 2072	25 × 1682	25 × 2312	25 × 2312	25 × 1512	25 × 1512	25 × 1512	25 × 612
Size of reduced feature vector matrix (**X**′)	10 × 2312	8 × 1682	9 × 2312	10 × 2312	11 × 1512	6 × 1512	10 × 1512	8 × 612
Training error	4.93e − 08	8.53e − 08	2.53e − 08	4.52e − 08	1.03e − 07	4.82e − 08	7.24e − 11	2.62e − 08

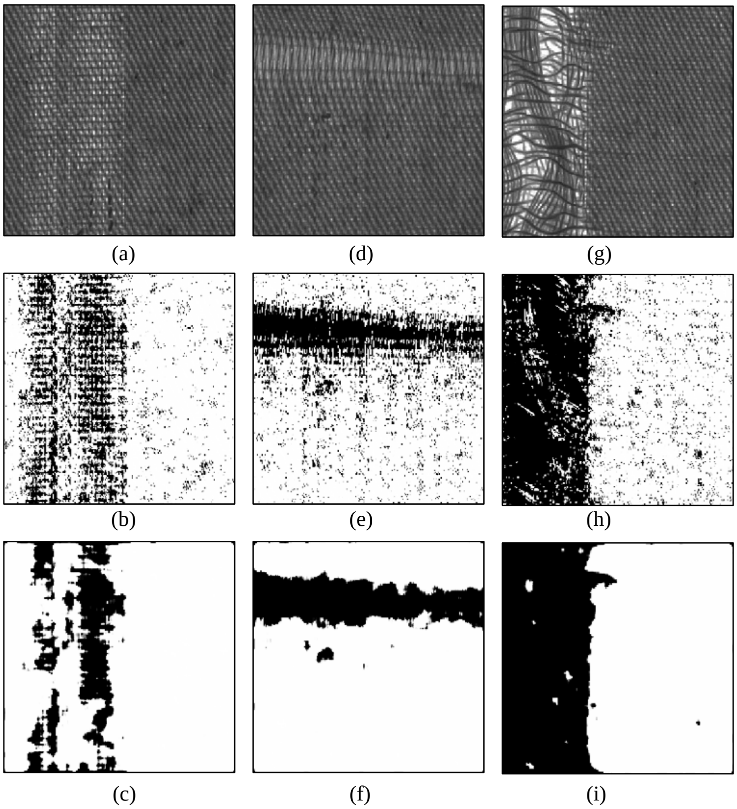

Fig. 6. Test of robustness of the segmentation scheme: fabric samples with defect *netting-multiplies* (1), *thin bar*, and *netting-multiplies* (2), in (a), (d) and (g), respectively (same as in Fig. 4); their corresponding segmented defects from the *mispick* trained FFN outputs in (b), (e), and (h), respectively; the corresponding segmented defects after postprocessing the FFN output images in (c), (f) and (i), respectively.

for the results summarized in Table 2. At every pixel, the FFN would require 130 (25 × 5 + 5) multiplications and 124 ((25 − 1) × 5 + (5 − 1)) additions. In addition, the network will also require 5 table lookups for the activation function of the hidden (5) units. Assuming that two instructions are required for a 256 word table lookup, every pixel will require 140 DSP instructions; 130 multiply and accumulate (MAC) instructions and 10 table lookup instructions. However, a *mispick* trained FFN with PCA will require 285 DSP instructions per pixel; 225 MAC for 25 × 9 PCA and (50 + 10) MAC for FFN. Assuming two frames per second is required, the 256 × 256 pixels image used in this work would require about 37.35 million DSP instructions per second to inspect 1.28 × 1.28 in^2 area of the fabric moving at about one foot per second. Thus the real-time implementation of this defect segmentation scheme using FFN would be computationally costly.

In this work, the detection of defects via segmentation has been investigated. Neural networks are among the best pattern classifiers and therefore have been used for the classification of defects [10,14,35]. Any extension of the proposed approach that could also classify the segmented defects into different categories would be highly desirable.

3. Defect detection using linear networks

Web inspection based on defect segmentation using FFN has been suggested in previous sections. Despite several efforts to reduce the computational time, real time web inspection using this approach will require additional DSP hardware, which will increase the cost of the inspection system. As the quest for quality increases, more and more industries aspire for automation in quality control. Therefore low-cost web inspection systems that can detect the defects and run on a conventional PC are in increasing demand. Such systems can only perform some limited real-time computations, *i.e.* without additional DSP hardware. One such possible solution is to reduce the inspection search space. In the textile industry, the weaving defects correspond at variations of the characteristics of interlaced yarns. Due to the nature of the weaving process, most of the fabric defects occur either in the vertical (warp) or horizontal (weft) direction [1]. Thus the search space can be reduced from 2-D image to 1-D signals, obtained from horizontal and vertical projections of pixel gray-level values. Kim et al. [20] have used such 1-D signals for the detection of fabric defects using Mexican hat wavelet at three scales. Linear neural networks can be used to model linear system with zero error, and nonlinear systems with minimum mean squared error. An efficient method of fabric defect detection using such linear networks has been developed. In next subsections, this approach is detailed and experimental results from the real fabric defects are presented.

3.1. Linear networks

A linear network with a single layer can classify an object in two categories, i.e. with-defect or without-defect. The linear network has a decision boundary that is determined by input vectors for which the network output is zero. The output of a single layer (single neuron) network having a weight vector **w**, input vector **x**, and the neuron bias λ is given by [36],

$$\varphi(k) = f(\mathbf{wx} + \lambda) = \sum_{q=1}^{N} w_{1q}\varphi(k-q) + \lambda. \tag{13}$$

The output in Eq. (13) resembles to the output of a finite impulse response (FIR) filter that acts as a *predictor*. The best linear network (*predictor*) will be the one that minimizes an error such as least squares. In a supervised training, a set of input (**x**) and target (**t**) vectors is provided for the desired network behavior. The object of training is to minimize the average of the sum of errors between the network ($\phi(k)$) and the target ($t(k)$) output, i.e.

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}\{t(k) - \varphi(k)\}^2. \tag{14}$$

This is achieved by the least mean square error (LMS) algorithm [37,38], which adjusts the bias and the weights of the linear network to minimize the mean square error (MSE). The LMS algorithm is well known for its computational simplicity. A five input, single neuron, linear layer network with its bias and weights computed from the supervised learning, will be used to predict the current values of a delayed input signal.

3.2. Methodology

The vibration free image $A(i,j)$ of fabric under inspection is used to obtain two 1-D projection signals, as shown in Fig. 7. The complete block diagram of the proposed defect detection method is shown in Fig. 8. The projection signals $p'_h(j)$ and $p'_v(i)$ are obtained in each of the horizontal and vertical directions by summing up the image pixel values along the columns and rows, respectively, as follows:

$$p'_h(j) = \sum_{i=1}^{R} A(i,j), \qquad p_h(j) = \frac{1}{\sigma_h}\{p'_h(j) - \mu_h\}, \tag{15}$$

$$p'_v(i) = \sum_{j=1}^{S} A(i,j), \qquad p_v(i) = \frac{1}{\sigma_v}\{p'_v(i) - \mu_v\}, \tag{16}$$

where μ_h and μ_v are the mean values of signal $p'_h(j)$ and $p'_v(i)$, respectively, and are used to make $p_h(j)$ and $p_v(i)$ zero mean. Similarly, σ_h and σ_v are the variance of signal $p'_h(j)$ and $p'_v(i)$, respectively, and are used to make $p_h(j)$ and $p_v(i)$ of unity variance. As shown in Fig. 8, a tapped delay line (D) is used to store the projection signal values

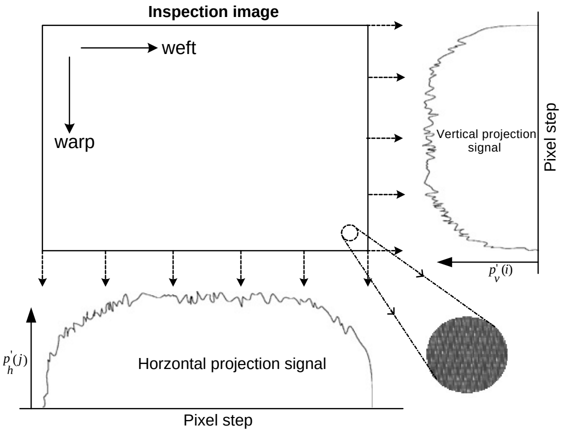

Fig. 7. Generation of two projection signals from the image under inspection.

while it passes from left to right. Past values of the signal $p_h(j)$ from the tapped delay line are used to predict the current value of signal, i.e. $\tilde{p}_h(j)$. Next, the absolute error of this prediction value $e_h(j)$ for every pixel step ($j=1,\ldots,S$) is computed.

$$e_h(j) = |\tilde{p}_h(j) - p_h(j)|, \quad j = 1,\ldots,S, \tag{17}$$

$$e_v(i) = |\tilde{p}_v(i) - p_v(i)|, \quad i = 1,\ldots,R. \tag{18}$$

The bias and the weights of the linear network layer used to compute $\tilde{p}_h(j)$ is computed from a reference or defect-free fabric $A_r(i,j)$. This reference fabric is used to generate two 1-D signals (defect-free, $p_{h_r}(j)$ and $p_{v_r}(i)$), as in Eqs. (15) and (16). Each of these 1-D signals is used to compute bias and weights of the network using least mean square error algorithm as described in Section 3.2. These bias and weights for horizontal $(\mathbf{w},\lambda)$ and vertical $(\mathbf{w}',\lambda')$ direction signals are computed before the beginning of the inspection process, and used for prediction during the inspection (Fig. 8).

The absolute errors in the prediction signals, i.e. $e_h(j)$ and $e_v(i)$ are subjected to the thresholding operation, and the resultant signal shows the defects and its approximate location. The thresholding value is again computed from the reference fabric before the beginning of inspection. The thresholding value for each of the horizontal and vertical directions is computed from $e_{h_r}(j)$ and $e_{v_r}(i)$) as

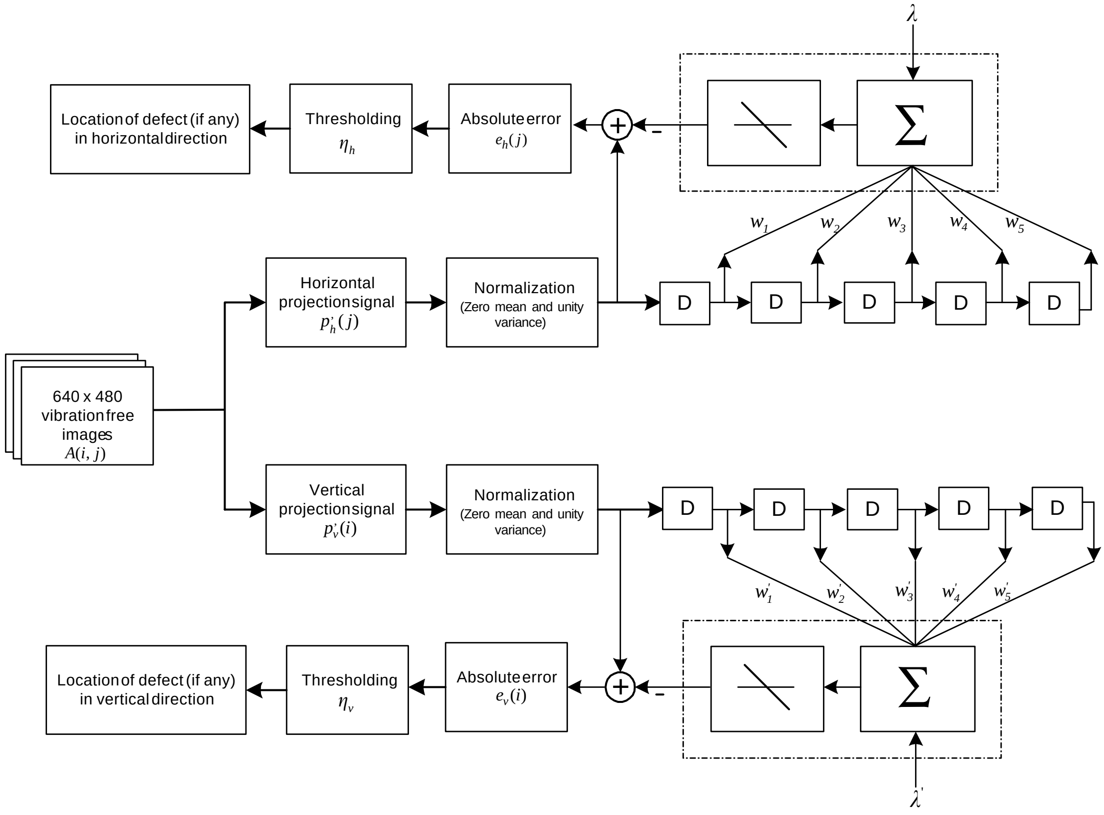
λ
Location of defect (if any) in horizontal direction
Thresholding η_h
Absolute error $e_h(j)$
+
-
Σ
w_1
w_2
w_3
w_4
w_5
Horizontal projection signal $p_h^{'}(j)$
Normalization (Zero mean and unity variance)
D
D
D
D
D
640 x 480 vibration free images $A(i, j)$
Vertical projection signal $p_v^{'}(i)$
Normalization (Zero mean and unity variance)
D
D
D
D
D
$w_1^{'}$
$w_2^{'}$
$w_3^{'}$
$w_4^{'}$
$w_5^{'}$
Location of defect (if any) in vertical direction
Thresholding η_v
Absolute error $e_v(i)$
+
-
Σ
$\lambda^{'}$

Fig. 8. Block diagram of the defect detection scheme based on linear neural networks.

follows [19]:

$$\eta_h = \arg\left(\max_{j\in Z}|e_{h_r}(j)|\right), \qquad \eta_v = \arg\left(\max_{i\in Z}|e_{v_r}(i)|\right) \qquad (19)$$

where ‘Z’ is a window for the observation of the absolute error signal. Thus the thresholded value η_h is the maximum amplitude of absolute error signal $e_{h_r}(j)$, within a window ‘Z’, obtained from the reference image. The window size is chosen to neglect the effect of large error in the prediction at the beginning. This is necessary as the error $e_{h_r}(j)$ is very large in the signal at the beginning (say $j = 1$), since the past values of signal $p_{h_r}(j)$ are assumed to be zero ($j < 1$).

3.3. Experimental setup and results

Low resolution inspection images of about 45 pixels per inch (640 pixels cover 14 in of fabric) are used for higher throughput, so that larger fabric area per frame can be processed. Using an Omnivision digital OV7110 camera with 640 × 480 pixels, vibration free BW images of fabric under inspection are acquired under backlighting. The signal $\tilde{p}_h(j)$ and $\tilde{p}_v(i)$ was predicted from the previous five (empirically determined) values of the corresponding signals. The window ‘Z’ was chosen to neglect the first five values of prediction error signal. The experimental results were excellent and some of these are in reproduced in Figs. 9–12.

Fig. 9 shows the fabric inspection image with the defect *slack-end*. Before the inspection of this image, the bias and weights for the linear layer in both directions are computed from a reference image. These bias and weights are used to predict the signal $\tilde{p}_h(j)$, and is shown in Fig. 9(b), where the original signal ($p_h(j)$) is plotted with the predicted signal ($\tilde{p}_h(j)$). The absolute error in the prediction signal $e_h(j)$ is shown in Fig. 9(c). The large prediction error due to the defect can be observed in this figure. The thresholding value η_h was computed as 0.0974 from the reference image. The signal in Fig. 9(c) is thresholded at this level, and the defects are shown in Fig. 9(d). Since the vertical signal $p_v(i)$ did not show any defect after thresholding, results from this signal are not reproduced here. Similar detection results from the fabric sample with defect *dirty-yarn*, from the horizontal direction signal $p_h(j)$ can be seen in Fig. 10. As seen from the results in Figs. 9 and 10 defects have been detected quite accurately.

Another fabric sample with defect *mispick* is shown in Fig. 11(a). The vertical projection signal $p_v(i)$ from this image is shown in Fig. 11(b). The absolute error signal $e_v(i)$ for this image exhibits two peaks, apart from the one at the beginning. The peak around the pixel step 183 can be clearly attributed to the observed high contrast defect in the image, however the defect causing the peak around pixel step 477 is difficult to observe from Fig. 11(a). Therefore, a zoom window near the boundary of this image is used to magnify the defect, and is reproduced in Fig. 11(a) to show the defect. For the image shown in Fig. 11(a) the thresholding value of $\eta_v = 0.6137$ was used. A similar detection of an-

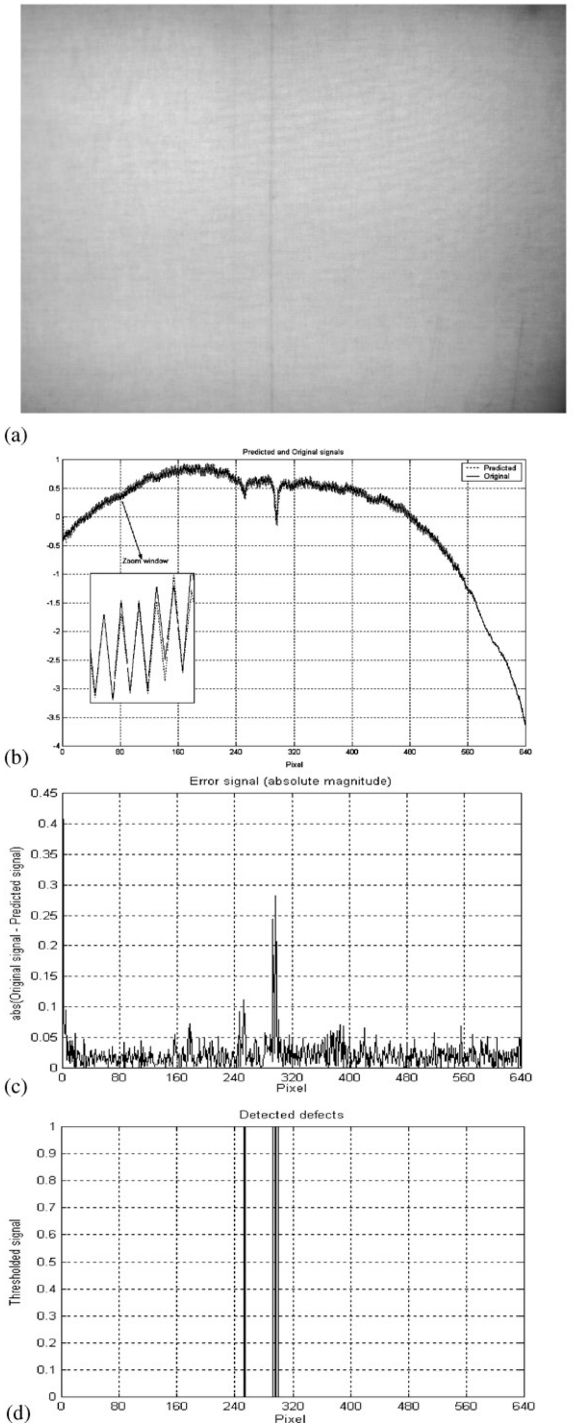

Fig. 9. (a) Fabric sample with defect *slack-end*, (b) horizontal projection signal and predicted signal, (c) absolute magnitude of error signal and (d) defects detected and their location.

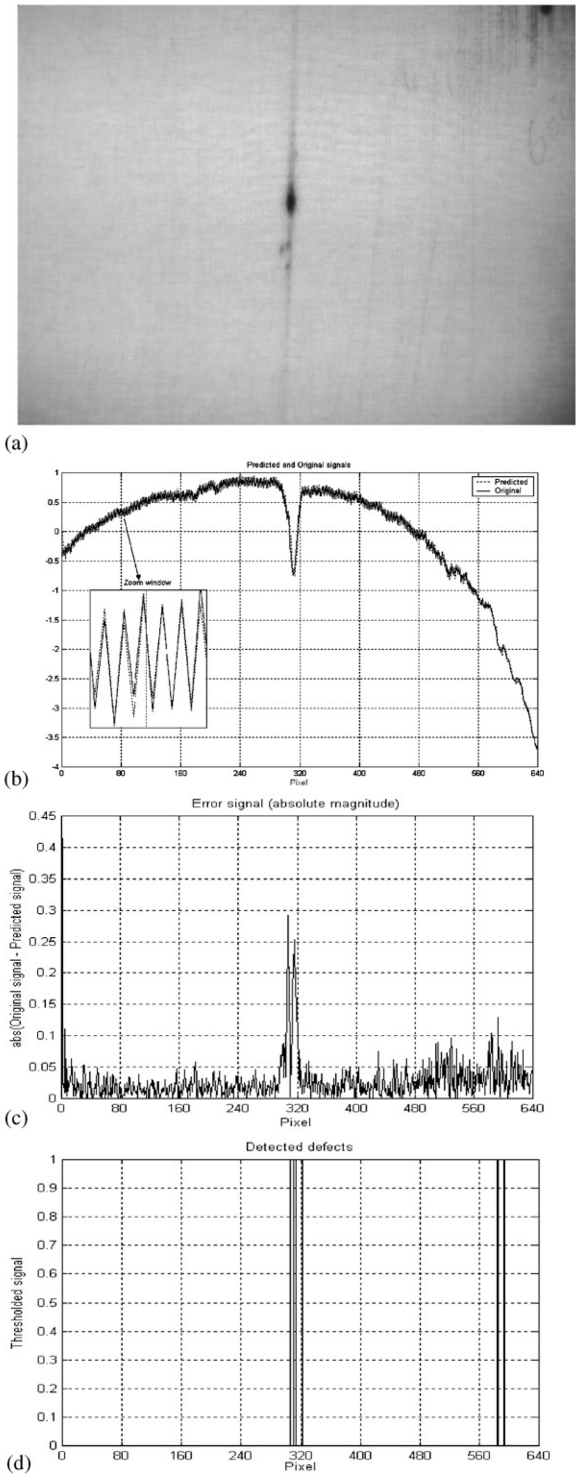

Fig. 10. (a) Fabric sample with defect *dirty-yarn* (b) horizontal projection signal and predicted signal, (c) absolute magnitude of error signal and (d) defects detected and their location.

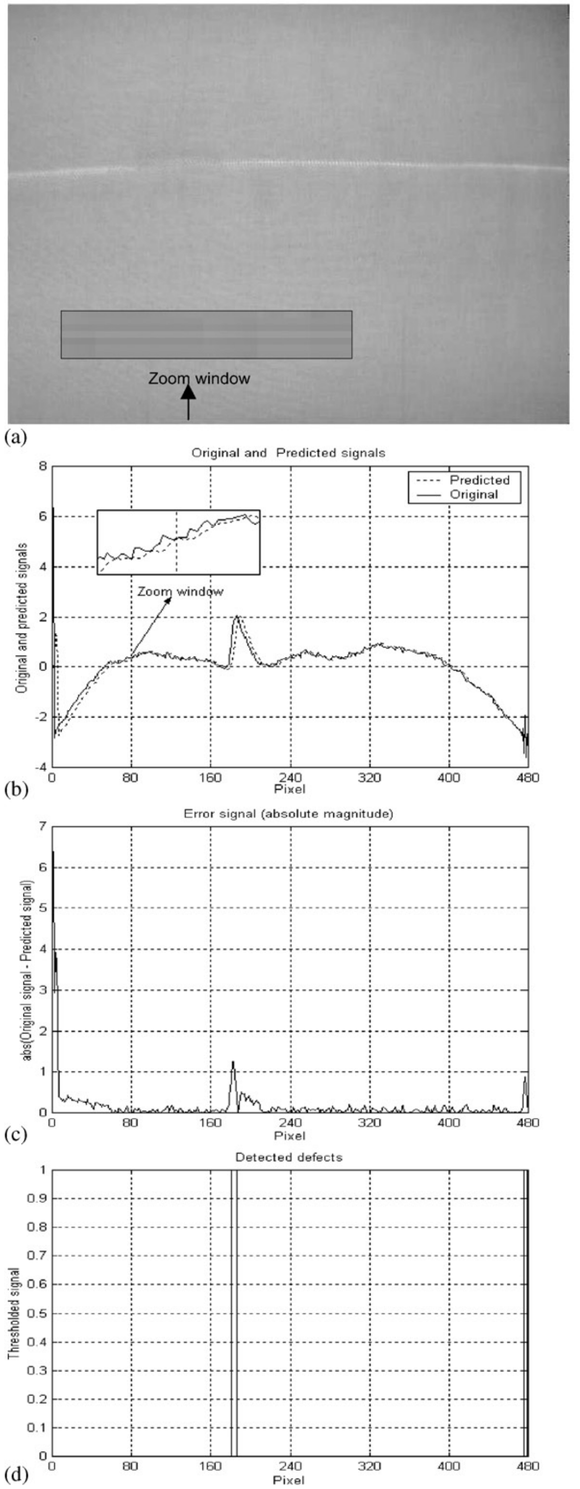

Fig. 11. (a) Fabric sample with defect *mispick*, (b) horizontal projection signal and predicted signal, (c) absolute magnitude of error signal and (d) defects detected and their location.

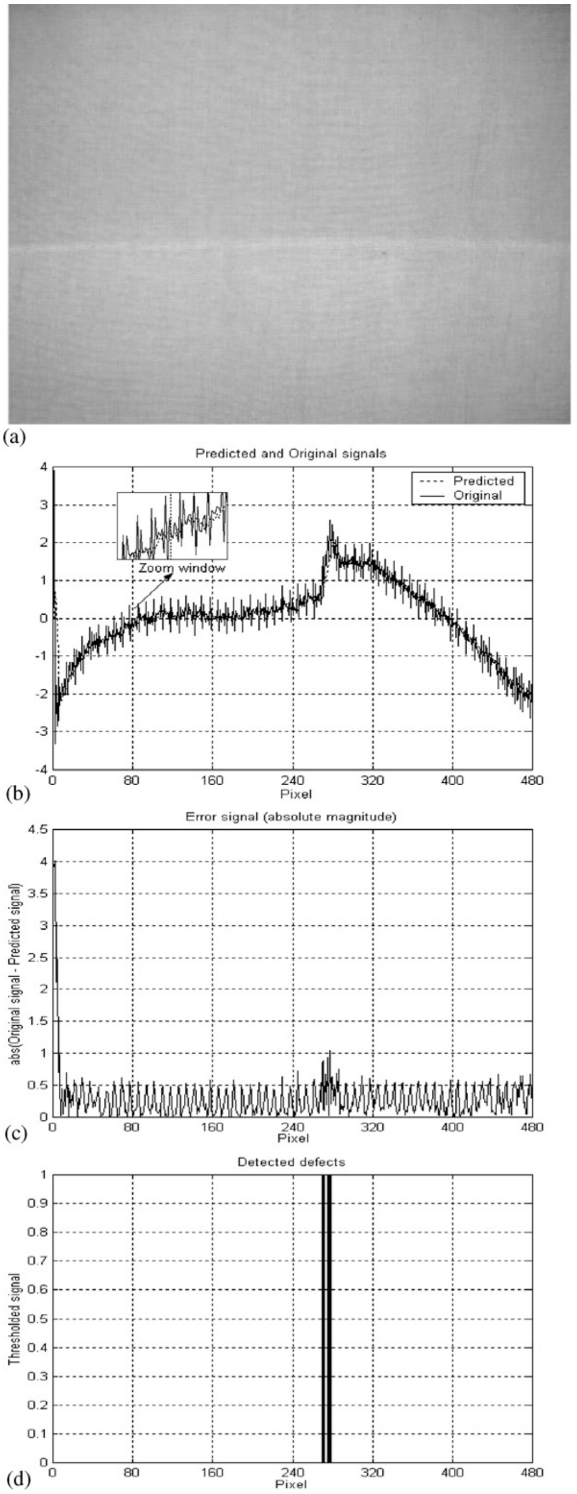

Fig. 12. (a) Fabric sample with defect *thin-bar*, (b) horizontal projection signal and predicted signal, (c) absolute magnitude of error signal and (d) defects detected and their location.

other defect *thin-bar* can be observed in Fig. 12. For the images shown in Figs. 11(a) and 12(a), there was no detection of any defect from the horizontal projection signal; therefore, the results from the signal $p_h(j)$ are not reproduced here. A number of other fabric defects such as *oil-stain*, *shuttle-mark*, *thin-bar*, etc. were successfully detected and this proved the robustness of the proposed defect detection scheme. Some fabric defects were simultaneously detected from both the projection signals $p_h(j)$ and $p_v(i)$, which was more useful in physical localization of defects in the fabric samples. The computational time of this algorithm as run on Pentium 450 MHz PC using a simple C program is 94 ms. Therefore real time inspection using this algorithm will not require any additional DSP hardware. Thus this algorithm offers a low-cost solution to the inspection problem.

3.4. Discussion

A single layer linear network used in this work finds the lowest error linear estimate for the nonlinear projection signals. Any major deviation from this linear fit model is assumed to be a defect. The linear model will be highly accurate as long as the nonlinear projection signal stays near a given operating point. This operating point mainly depends on the texture of fabric under inspection and the illumination conditions. The experimental results with the adaptive predictor, in which the bias and weights of network were dynamically computed (using Widrow–Hoff learning rule [39] with 0.1 as the learning rate) so as to reduce the prediction error, were not satisfactory. This was due to the fact that the adaptive predictor catches up with the varying signal from the defect with very low prediction error, and thus makes it very difficult to detect defects (especially when defects causes a slow variation in $p_h(j)$ or $p_v(i)$ signal). The creases (wrinkle) present in the fabric under inspection are detected as defects and therefore generates false alarms. Therefore, the imaging system has to be adjusted in such a way so as to avoid wrinkles in the low-resolution images. When the contrast associated with the defect is low, the detection of defects becomes more difficult because the projection signals fail to register the significant change is amplitude, required to detect the defect.

Another observation that can be made from the experimental results is that the distortions introduced in the image under inspection due to the non-uniform illumination (Figs. 9–12), does not have any effect on the detection of defects, as long as the contrast associated with a defect does not change. The low frequency distortion signal, superimposed on high frequency signal (from projection), has not generated any problem (i.e. false alarm) in the detection of defects even at the edges (Figs. 9 and 10). A potential improvement in the detection rate of this algorithm can be achieved by block processing in each of the two directions. The inspection image can be divided into several rectangular overlapping blocks and defects can be detected from the projection signals in each of these blocks (using prediction error computed from the bias and weights corresponding to each of blocks). However this will require more computations, but it can improve the detection rate and localization of defects in the fabric.

4. Conclusions

In this paper, a new approach for fabric defect segmentation using FFN has been demonstrated. Several attempts to reduce the computational requirements have yielded successful results. The tests conducted on different types of defect and different styles of fabrics have shown that this technique is efficient and robust for a variety of fabric defects. A low-cost web inspection system based on linear neural networks has been also proposed in this paper. The performance of this approach has been evaluated on real fabric samples. The results have shown that this method is quite successful, and offers a low-cost, single PC based solution for fabric defect detection.

The approach presented in Section 3 offers a tradeoff between performance and cost of inspection system. This tradeoff is reasonable since the majority of defects have been detected by this method. Some of the small sized defects, which cannot be seen in the low-resolution inspection images, cannot be detected by this method. Detection of such defects requires high-resolution inspection image. Therefore the segmentation approach, such as presented in Section 2, which also provides accurate localization of the dimension and location of defects, is recommended.

Recently, Chetverikov et al. [25] have detailed a new method for texture defect detection using two fundamental structural properties of texture, i.e. structural regularity and local anisotropy. The authors have quantified these two properties and used each of them in two separate/corresponding approaches to declare local texture defects. The defect segmentation method evaluated in this paper uses structural arrangement of neighboring gray levels as a texture feature, which can be interpreted as simplified combination of (i) feature of regularity, i.e. periodicity of gray levels and (ii) dominating pattern orientation used in Ref. [25]. This method [25] is more practical and does not require any training from defect samples, and is hence suitable for the inspection of defects with the dynamic defect population. Therefore, it is suspected that the method investigated in Section 2 does not yet compete in performance with the method in Ref. [25]. In this paper the defect detection results only from the real fabric samples have been demonstrated. However, the similar results can also be obtained for detection of defects in other textured materials such as paper, plastic, wood or steel-rolls.

References

[1] H. Sari-Sarraf, J.S. Goddard, Vision systems for on-loom fabric inspection, IEEE Trans. Ind. Appl. 35 (1999) 1252–1259.

[2] D. Brzakovic, H. Sari-Sarraf, Automated inspection of non-woven web materials: a case study, Proc. SPIE 2183 (1994) 214–222.

[3] R.W. Conners, C.W. McMillin, K. Lin, R.E. Vasquez-Espinosa, Identifying and locating surface defects in wood: part of an automated lumber processing system, IEEE Trans. Pattern Anal. Mach. Intell. 5 (1983) 573–583.

[4] J. Laitinen, Image quality in automated visual web inspection, Proc. SPIE 3029 (1997) 78–89.

[5] J.W. Roberts, S.D. Rose, G.A. Jullian, G.A. Nichols, L. Jenkins, P. Chamberlain, S.G. Maroscher, PC based real-time defect imaging system for high speed web inspection, Proc. SPIE 1907 (1993) 164–176.

[6] L. Siew, R.M. Hodgson, E.J. Wood, Texture measures for carpet wear assessment, IEEE Trans. Pattern Anal. Mach. Intell. 10 (1988) 92–105.

[7] F.S. Cohen, Z. Fau, S. Attali, Automated inspection of textile fabrics using textured models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 803–808.

[8] J.G. Campbell, C. Fraley, F. Murtagh, A.E. Raftery, Linear flaw detection in woven textiles using model-based clustering, Pattern Recognition Lett. 18 (1997) 1539–1548.

[9] F. Ade, Application of principal component analysis to the inspection of industrial goods, Proc. SPIE 397 (1983) 216–223.

[10] I.-S. Tsai, Ch-H. Lin, J.-J. Lin, Applying an artificial neural network to pattern recognition, Text. Res. J. 65 (1995) 123–130.

[11] X.F. Zhang, R.R. Bresee, Fabric defect detection and classification using image analysis, Text. Res. J. 65 (1) (1995) 1–9.

[12] M. Unser, F. Ade, Feature extraction and decision procedure for automated inspection of textured materials, Pattern Recognition Lett. 2 (1984) 181–191.

[13] A. Kumar, Automated defect detection in textured materials, Ph.D. Thesis, Department of Electrical and Electronic Engineering, The University of Hong Kong, May 2001.

[14] E.J. Wood, Applying Fourier and associated transforms to pattern characterization in textiles, Text. Res. J. (1990) 212–220.

[15] L.H. Hoffer, F. Francini, B. Tiribilli, G. Longobardi, Neural networks for the optical recognition of defects in cloth, Opt. Eng. 35 (1996) 3183–3190.

[16] S.A. Hosseini Ravindi, Fourier transform analysis of plain weave fabric appearance, Text. Res. J. 65 (1995) 676–683.

[17] W.J. Jasper, S.J. Garnier, H. Potapalli, Texture characterization and defect detection using adaptive wavelets, Opt. Eng. 35 (1996) 3140–3149.

[18] J. Escofet, R. Navarro, M.S. Millan, J. Pladelloreans, Detection of local defects in textiles webs using Gabor filters, Opt. Eng. 37 (1998) 2297–2307.

[19] A. Kumar, G. Pang, Fabric defect segmentation using multichannel blob detectors, Opt. Eng. 39 (12) (2000) 3176–3190.

[20] S. Kim, M.H. Lee, K.B. Woo, Wavelet Analysis to defects detection in weaving processes, in: Proceedings of IEEE International Symposium on Industrial Electronics, Vol. 3, July 1999, pp. 1406–1409.

[21] J.G. Vachtsevanos, M. Mufti, J.L. Dorrity, Method and apparatus for analyzing an image to detect and identify defects, US Patent No. 5, 815,198, September 1998.

[22] M. Tuceyran, A.K. Jain, in: C.H. Chan, L.F. Pau, P.S.P. Wang (Eds.), Texture Analysis-Handbook of Pattern Recognition and Computer Vision, World Scientific, Singapore, 1993, pp. 235–276, (Chapter 2).

[23] A. Kumar, G. Pang, Defect detection in textured materials using optimized filters, IEEE Trans. Syst. Man Cybernet. Part B: Cybernet. 32 (5) (2002) 553–570.

[24] D. Chetverikov, Pattern regularity as a visual key, Image Vision Comput. 18 (2000) 975–986.

[25] D. Chetverikov, A. Hanbury, Finding defects in texture using regularity and local orientation, Pattern Recognition 35 (2002) 2165–2180.

[26] C.M. Bishop, Neural networks for pattern recognition, Clarendon Press, Oxford, 1995.

[27] R.M. Harlick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybernet. 3 (1973) 610–621.

[28] A.K. Jain, K. Karu, Learning texture discrimination masks, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1996) 195–205.

[29] J. Mao, A.K. Jain, Texture classification and segmentation using multiresolution simultaneous autoregressive models, Pattern Recognition 25 (2) (1992) 173–188.

[30] I.T. Jolliffe, Principal Component Analysis, Springer, New York, 1986.

[31] J.P. Reilly, Matrix computations for signal processing, EE 731 Lecture Notes (3), Mc Master University, also available on http://www.crl.mcmaster.ca/People/Faculty/Reilly/reilly.html, February 2000.

[32] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd Edition, John Hopkins University Press, 1996.

[33] M.T. Hagan, M. Menhaj, Training feed-forward networks with the Marquardt algorithm, IEEE Trans. Neural Networks 5 (1994) 989–993.

[34] Timothy Masters, Advanced Algorithms for Neural Networks: A C++ Sourcebook, Wiley, New York, 1995.

[35] W.J. Staszewski, K. Worden, Classification of faults in gearboxes—pre-processing algorithms and neural networks, Neural Comput. Appl. 5 (1997) 160–183.

[36] H. Demuth, M. Beale, Neural Network Toolbox Users Guide, The Mathworks, Inc., 1998.

[37] J.P. Reilly, Matrix computations for signal processing, EE 731 Lecture Notes (7), Mc Master University, also available on http://www.crl.mcmaster.ca/People/Faculty/Reilly/reilly.html, February 1998.

[38] C.L. Lawson, R.J. Hanson, Solving Least Squares Problems, SIAM, Philadelphia, 1995.

[39] B. Widrow, S.D. Sterns, Adaptive Signal Processing, Prentice-Hall, New York, 1985.

About the Author—AJAY KUMAR received the Ph.D. degree from The University of Hong Kong in May 2001.
He was with the Indian Institute of Technology (IIT), Kanpur, as Junior Research Fellow and at IIT Delhi as Senior Scientific Officer before joining Indian Railways. He joined the Indian Railway Service of Signal Engineers (IRSSE) in 1993 and worked as Assistant Signal and Telecom Engineer. He worked as Project Engineer at IIT (Kanpur) during 1996–97 and as Assistant Professor at NIST, Berhampur, India, from September 1997 to September 1998. He was a Research Associate with The University of Hong Kong from December 1998 to August 1999. He completed his doctoral research at The University of Hong Kong in a record time of 21 months (September 1999–May 2001). Currently he is engaged in postdoctoral research in the Department of Computer Science, Hong Kong University of Science and Technology. His research interests include pattern recognition with the emphasis on biometrics and defect detection using wavelets, general texture analysis, neural networks and support vector machines.