doi: 10.1111/cote.12394

Automatic fabric defect detection using a deep convolutional neural network

Jun-Feng Jing*, Hao Ma and Huan-Huan Zhang

School of Electronics and Information, Xi'an Polytechnic University, Xi'an, 710048, China
Email: jingjunfeng0718@sina.com

Received: 16 May 2018; Accepted: 4 February 2019

Fabric defect detection plays an important role in the textile production process, but there are still some challenges in detecting defects rapidly and accurately. In this paper, we propose a powerful detection method for automatic fabric defect detection using a deep convolutional neural network (CNN). It consists of three main steps. First, the fabric image is decomposed into local patches and each local patch is labelled. Then the labelled patches are transmitted to the pretrained deep CNN for transfer learning. Finally, defects are detected during the inspection phase by sliding over the whole image using the trained model, and the category and position of each defect is obtained. The proposed method is validated on two public and one self-made fabric database. The experimental results demonstrate that our method significantly outperforms selected state-of-the art methods in terms of both quality and robustness.

Introduction

Because of yarn problems, improper operation and over-stretch, various defects can be formed on the surface of fabrics. To date, the textile industry has defined more than 70 types of fabric defects [1]. Figure 1 shows six kinds of common defects. According to statistics, if there are defects in fabric then the price is reduced by 45–65% [2]. Therefore, fabric defect detection is an essential step for quality control in textile manufacturing. The traditional method is to detect defects via human vision, which can help workers to repair minor defects immediately. But the efficiency of manual detection can reduce gradually with increases in working time. Thus, it is necessary to develop an automatic inspection system for fabric defects to improve the quality of fabric, and to reduce human labour costs and errors.

Over the years, in targeting fabrics with a homogeneous texture structure [3] such as plain weave [4], twill [5] and leather fabric [6], researchers have proposed many methods, including auto-correlation function (AF) [7,8], local binary pattern (LBP) [9], Fourier transform (FT) [10,11], wavelet transform (WT) [5,12] and neural networks [13,14], to solve these fabric defect detection problems. These methods were used to detect defects at the image level, so it is difficult to locate the defects accurately. Therefore, they cannot be extended to other types of fabrics. Recently, some other algorithms based on local image level [15,16] have been proposed, which use the minimum unit as the basic operation object to extract image features. The similarity measurement between the images being detected and the template image are computed, and then the threshold value is given to achieve the defect detection. However, this often takes more time because of the process of computing AF.

In this study, we introduce a novel defect detection algorithm which can deal with different types of defects. Our method does not consider the original image as a basic operation unit [17]. Instead, the local image patches are used in the training phase, while the whole fabric image is used during the test phase [patches training and image testing (PTIT)]. First, the Mixed National Institute of Standards and Technology (MNIST) [18] dataset is used to pretrain the proposed model and save the model parameters, which are used as initialisation parameters of our network to avoid overfitting caused by small-scale fabric samples. Second, the fabric image is decomposed into multiple repeat units (RUs) and the label of defect class is given manually. The pretrained model parameters are loaded and the labelled fabric data are fed into the network to fine-tune the parameters, thereby accelerating and optimising the learning efficiency of the model. Finally, defects are detected during image inspection by sliding on the image to test local patches using the learned model. Experimental results showed that our method is better than the selected state-of-the-art methods.

The remainder of this paper is organised as follows. Next, we present a brief literature review of fabric defect detection methods. Then we propose our method for automatic defect detection, which includes the automatic calculating of the fabric surface period and the proposed network model. This is followed by a description of the three datasets that are used for evaluation. Finally, results and discussion are followed by conclusions.

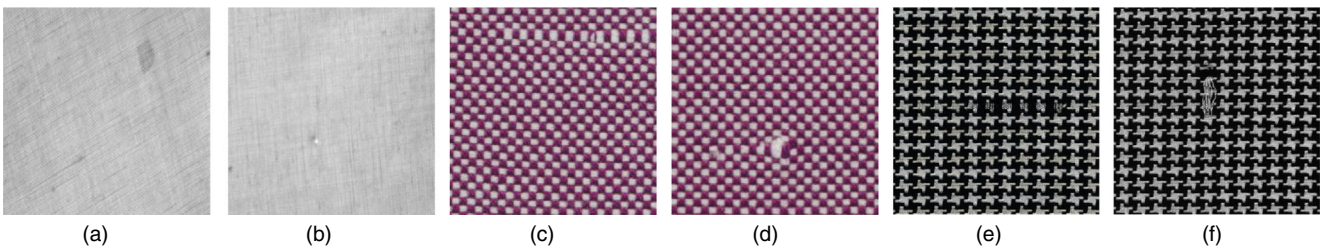

Figure 1 There are six kinds of common types of defects: (a) Stain, (b) Hole, (c) Carrying, (d) Knot, (e) BrokenEnd, and (f) NettingMultiple [Colour figure can be viewed at wileyonlinelibrary.com]

Review of previous work

Most of the existing work on fabric inspection is mainly focused on homogeneous fabrics [3,19], including plain and twill fabrics. These methods can be classified into four main categories: (i) statistical; (ii) spectral; (iii) model-based; and (iv) learning-based approaches.

In statistical approaches, the AF and the co-occurrence matrix (CM) [15,20] have been successfully applied to detect defects. Zhu *et al.* [15] combined the AF and the grey level co-occurrence matrix (GLCM) methods to detect yarn-dyed fabric defects. The AF helps to determine the size of the template image. GLCM can represent characteristics of an image, such as the contrast, correlation, energy and entropy. Latif-Amet *et al.* [20] have applied CM analysis in the wavelet domain to detect defects. However, these methods are time-consuming.

Spectral approaches are widely used to defect detection, and these include the FT [10,11] and the WT [5,21–23]. Wood [8] applied Fourier masks to restrain low- or high-frequency components in an image, enhancing the texture in the image and measuring the roughness of the defective carpet. Guo *et al.* [24] and Campbell *et al.* [25] used the FT to detect defects in woven fabrics and saliency detection, respectively. The disadvantage of using FT is the lack of local information in the spatial domain and the insensitivity towards small defects. Contrary to FT, Gabor filters and the WT utilise a spatial frequency analysis, which realises the detection of local defects [16]. Ngan *et al.* [23] used the WT to automatically detect defects on patterned fabric with an accuracy of 96.7%.

Model-based methods are used to solve the defect detection problem by assuming that the texture obeys a particular distribution model and that the model's parameters are estimated [16,26]. Yapi *et al.* [16] and Allili *et al.* [27] divided the image into elementary repetitive units and simulated the distribution of redundant contourlet transform (RCT) coefficients using a finite mixture of a generalised Gaussian model. These methods can deal with various types of textile fabrics. But they are also more time-consuming. Cohen *et al.* [26] used Gaussian Markov fields (GMFs) to model defect-free texture on fabric images; however, they are inefficient in detecting minor defects.

Learning-based approaches are also popular in detecting defects, using labelled samples to train classifiers that distinguish between defective and non-defective samples. Convolutional networks have good fault tolerance, parallel processing capabilities, generalisation capabilities, and self-learning capabilities that can handle complex environmental information issues. Since the early 2000s, with the rapid development of big data and artificial intelligence, convolutional networks have been applied with great success to the detection, segmentation [28] and recognition [29] of objects and regions in images, especially in tasks with a large number of labelled samples, such as rail surfaces [30], industrial images [31], traffic sign recognition [32], face detection [33] and natural language processing [34–37]. Jing *et al.* [38] established a deep convolution neural network based on the characteristics of yarn-dyed fabric to achieve defect classification, and then used the Meanshift algorithm to segment defects and locate defect locations. This method is good at dealing with yarn-dyed fabric of complex textures, and the detection time of the algorithm is relatively short; furthermore, it can realise real-time defect detection. Ren *et al.* [39] used feature transferring to extract the features of the images for the detection tasks with small samples. The Felzenszwalb's segmentation [40] is applied to accurately locate the size and location of defects in the heatmaps, but it incurs a huge computation time to generate heatmaps.

PTIP method

Here, details of the PTIP method are presented. Our proposed method consists of three main basic steps. In the first step, we use the MNIST [18] dataset to pretrain our proposed model and save the parameters (weight, bias) of the optimal model as the initial parameters of the network. The second step consists of decomposing the image into local RUs, marking the corresponding defect categories, and fine-tuning the parameters of the model. In the final step, the trained model slides on the original image to detect each block.

Automatically calculating the patch size

By using the features of distance matching function (DMF), the period of the fabric image can be accurately and automatically calculated. We set the size of the period to patch size and decompose the image into multiple patches according to patch size, which can increase the training dataset at multiple levels and improve the precision of defect detection. Oh *et al.* [17] put forward a one-dimensional DMF $\lambda(\delta)$ to calculate the periodic distance, which can be calculated with DMF as Eqn (1):

$$\lambda(\delta) = \sum_{i=1}^{N-\delta} [f(i) - f(i+\delta)]^2 \tag{1}$$

where $\delta \in (0, N-1)$ represents periodic distance of one-dimensional function $\lambda(\delta)$ and N represents the length of the signal. When parameter δ is the period of function $f(i)$, the difference between function $f(i)$ and $f(i + \delta)$ is at its smallest.

Fabric can be considered as a two-dimensional (2D) function [1]. The size of function $f(x, y)$ corresponds to $M \times N$ size. The DMF of row r and column c are defined as follows:

$$\lambda_r(\delta) = \sum_{i=1}^{N-\delta} [f(r,i) - f(r,i+\delta)]^2 \tag{2}$$

$$\lambda_c(\delta) = \sum_{i=1}^{M-\delta} [f(i,c) - f(i+\delta,c)]^2 \tag{3}$$

where $\delta \in (0, N-1)$ represents the periodic distance of 2D function $\lambda(\delta)$.

For 2D image function $f(x, y)$, the resultant difference between δ and $\delta + 1$ of DMF is defined as the first forward difference $\Delta\Lambda_r(\delta)$. The resultant difference between δ and $\delta-1$ of DMF is defined as the first forward difference $\Delta\Lambda_r(\delta-1)$. The first forward difference of DMF in the vertical direction is defined as follows [41]:

$$\Delta\Lambda_r(\delta) = \Lambda_r(\delta+1) - \Lambda_r(\delta) \tag{4}$$

$$\Delta\Lambda_r(\delta-1) = \Lambda_r(\delta) - \Lambda_r(\delta-1) \tag{5}$$

The second forward difference of the fabric image in the vertical direction is defined as follows:

$$\Delta^2\Lambda_r(\delta) = \Delta\Lambda_r(\delta) - \Delta\Lambda_r(\delta-1) \tag{6}$$

The value of the independent parameter δ is considered as the period of the fabric in the vertical direction when the maximum value of the second forward difference is obtained.

Similarly, the first forward difference and the second forward difference of DMF in the horizontal direction are given in Eqns (7–9). The period of the fabric (patch size) can be accurately calculated using the second forward difference of DMF, and the result is shown in Figure 2.

$$\Delta\Lambda_c(\delta) = \Lambda_c(\delta+1) - \Lambda_c(\delta) \tag{7}$$

$$\Delta\Lambda_c(\delta-1) = \Lambda_c(\delta) - \Lambda_c(\delta-1) \tag{8}$$

$$\Delta^2\Lambda_c(\delta) = \Delta\Lambda_c(\delta) - \Delta\Lambda_c(\delta-1) \tag{9}$$

Manual labelling category

After the original image is decomposed into numerous patches, it is necessary to manually separate the defective and defect-free patches. For example, a defective fabric image produces at least two kinds of patches after decomposition, namely defective patches and defect-free patches. And the number of defect-free patches is much larger than the number of defective patches, as shown in Figure 3. Therefore, the category of the input image cannot be used directly as the output category of the convolution network; on the contrary, the category of the convolution network output is determined according to the category in the patches.

As shown in Figure 4b, the Knots defect image contains a defect patch of the Carrying type. The purpose of manual sorting is to classify the same defect patches in different original defect images into one class, as shown in the blue box in Figure 4; the yellow box is the Knots class. To prevent overfitting, following [42–44], we augment the number of defective patches through 90°, 180° and 270° rotations of the existing data. The patches are shuffled randomly and split into training (70%) and testing (30%). In each category, the ratio of non-defective sub-blocks to defective sub-blocks is 3:2. To prevent overfitting, one strategy is to add the dropout layer and set the corresponding weight values of the neurons to 0 with a predefined specific probability. With the dropout strategy, the performance of the model can be enhanced by performing a more effective training process for more accurate predictions.

Convolutional neural network

Convolutional neural networks (CNNs) are multilayer neural networks which are skilled in dealing with related machine learning problems, especially large-scale image classification tasks. During the training process of CNNs, the features of the input images can be extracted layer by layer, and the features information of the defect is finally obtained via feature fusion. A typical CNN normally consists of three parts: the convolutional layer, the pooling layer, and the fully connected layer.

Convolution is the core of the CNN, and uses multiple convolution kernels to traverse the input image to obtain the corresponding feature map. Supposing that the weight of filter is W, s denotes the stride of filter, $p \times q$ is the size of filter, b is the bias of filter, x represents the patch of size $m \times n$ on the input image, σ is the activation function, and the size of the input image is $M \times N$. Then the convolution operation is defined as:

$$f = \sigma(Wx + b) \tag{10}$$

After the convolution operation, the size of the output image is $\left[\left[\frac{M-p}{s}\right]+1\right] \times \left[\left[\frac{N-q}{s}\right]+1\right]$, where ⌊ ⌋ is the ceiling function.

The pooling layer is usually applied to compress the feature map and to extract the main features, which simplifies the complexity of network computation and

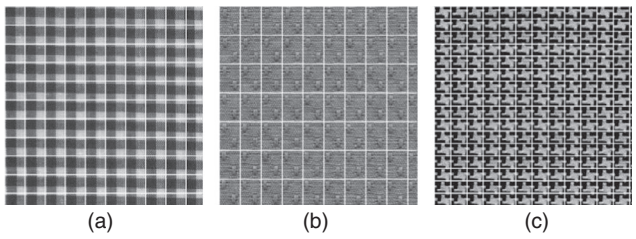

Figure 2 Using the second forward difference of DMF to calculate the patch size of fabric images: (a) box-patterned fabric, (b) dot-patterned fabric, and (c) star-patterned fabric

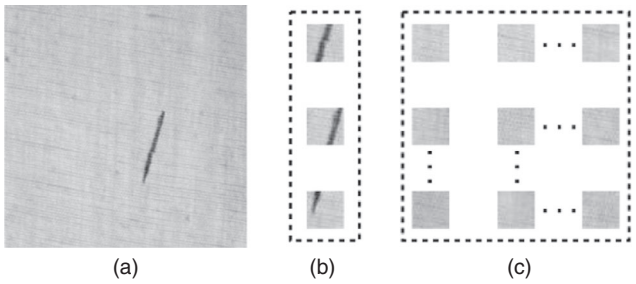

Figure 3 Manual labelling category diagram: (a) original image; (b) three defective patches in the original image; and (c) defect-free patches in the original image

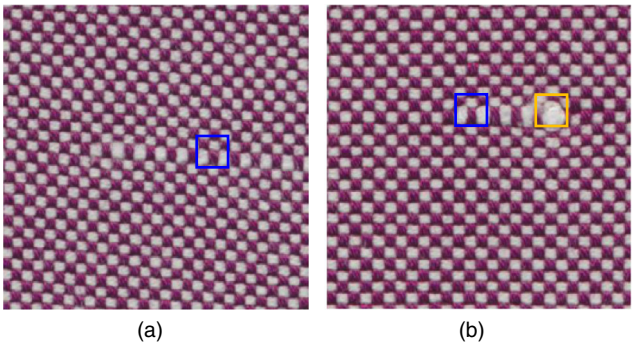

Figure 4 Manually filtered categories of patches: (a) Carrying and (b) Knots [Colour figure can be viewed at wileyonlinelibrary.com]

avoids overfitting problems. There are generally two kinds of pooling operation, one is Avy pooling, and the other is Max pooling. Max pooling is extensively used because it can reduce the error caused by mean shift.

The fully connected layer connects all the features and sends the output values to the classifier, such as the softmax classifier. A typical deep CNN (DCNN) usually consists of several rounds of convolutional and pooling layers, finally followed by fully connected layers.

Network model

DCNNs are good at learning from high dimensional data, but often require huge amounts of labelled examples for training. For tasks with fewer labelled samples, training DCNNs from scratch can easily result in model overfitting. However, several researchers trained DCNNs with large-scale labelled data and applied trained weights as the feature extractors for smaller datasets in another field. Transfer learning can greatly improve the learning performance, especially when only a few datasets are available in a target domain [45,46].

The fabric has a strong structure of texture on the surface. If there is a defect on the fabric, it will break the structure of the texture. Compared with other methods, our method does not directly use the original image as the input. Instead, we use multiple RUs along the inherent period of the fabric surface by dividing the fabric image. Therefore, the network model proposed in this paper is similar to the LeNet-5 [47] model, but the number of feature maps per layer is increased so that the model can learn enough features and use two full connected layers to reduce the trainable parameters in the model.

The proposed model structure is shown in Figure 5. It contains five learned layers, including three convolutional and two fully connected layers. In this model, the input images are of size 28 × 28 pixels with grey channels. The three convolution layers are filtered by a 3 × 3 kernel to achieve the goal of increasing non-linearity and reducing parameters. Each convolution layer has 16, 32 and 63 kernels, respectively.

With the deepening of the network, the input distribution of each layer gradually shifts to the upper and lower ends of the value range of the non-linear function. In backward propagation, it may cause the gradient of the low-level neural network to disappear. The batch normalisation (BN) [48] layer is added between the convolution layer. Its function is to keep the input of each layer of neural networks at the same distribution during deep neural network training, so that the gradient becomes larger to avoid the problem of gradient disappearance.

The ReLU activation function is used to enhance the non-linearity of the network and avoid the problem of subsequent gradient disappearance. The ReLU function is used as the activation function after each convolution layer, except for the fifth fully connected layer in this model. The fourth layer is the fully connected layer, and its neurons are connected to all the neurons in the previous layer with 2048 neurons. The fifth fully connected layer has as many neurons as the number of classes in the MNIST dataset used in the pretraining process. Pretraining the model on the MNIST dataset, the classification accuracy is 98.43%, and the trained parameters are set as initialisation parameters for training fabric data.

Dataset

A wide range of fabrics data with different textures were selected to demonstrate the accuracy of our method. These

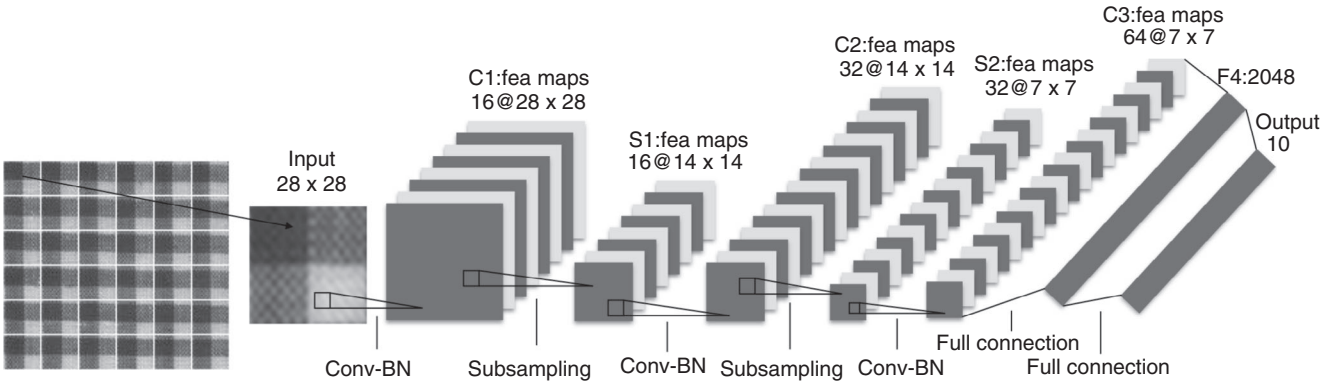

Figure 5 Architecture of a deep convolutional neural network. Each layer is based on four processing steps, namely, convolution (Conv), batch normalisation (BN), non-linear activation (not shown) and feature pooling for dimension reduction

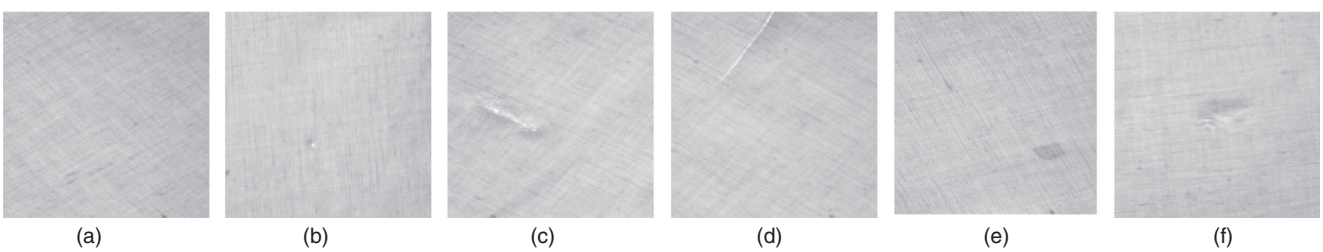

Figure 6 Some typical images in the TILDA database: (a) Non-defective, (b) Holes, (c) Carrying, (d) Scratch, (e) Stain, and (f) Knots

data include the white grey fabric in the TILDA database, homemade dark red fabric, and patterned texture fabric.

TILDA database

TILDA is a textile texture database, which includes eight kinds of representative textile categories. Based on the analysis of textile atlases, seven error classes and a correct class are defined. In the TILDA database, we chose 300 texture images which are divided into six classes (Non-defective, Holes, Carrying, Scratch, Stain, and Knots), and each class contains 50 samples. The size of the original image is 256 × 256 pixels, and some typical images are shown in Figure 6.

Dark red fabric

The dark red fabric is created by using the yarn-dyed fabric images collected from Guangdong Esquel Textiles (Guangdong Sheng, China). By manually sorting four classes (Carrying, Thin Bar, Knots, and Fuzz Balls), each class contains 30 samples. The representative samples for each class are shown in Figure 7.

Patterned texture fabric (regular patterned fabric)

The fabrics (box-patterned fabric, dot-patterned fabric, and star-patterned fabric) in the patterned fabric database consist of three different textures, which contain six types of fabric defect. Figure 8 shows some typical defect images. This fabric database is derived from Henry Y. T. Ngan (Industrial Automation Research Laboratory, Dept. of Electrical and Electronic Engineering, University of Hong Kong).

Experimental results

To evaluate the performance of our method, we compared the results of manual design features, transfer learning, and the proposed method for the three fabric databases. Methods of manually designed features include GLCM [15], LBP [9] and RCT [16]. Models for transfer learning include LeNet-5, AlexNet [49] and VGG16 [50]. The processor is Intel (R) Core (TM) i5 – 4460 CPU @ 3.20 GHz, RAM 8.00 GB. The graphics card type is NVIDIA GeForce GTX 745. MATLAB2017b software is used to implement the proposed algorithm.

The GLCM contains many features such as energy, contrast, correlation, entropy, and inverse difference moment. According to [15], the greyscale of the image (Ng), inter-pixel distance (d) and inter-pixel orientation (θ) are Ng = 16, $d = 3$ and $\theta = 0°$, respectively. LBP is an operator used to describe the local texture features of images. It has the merit of rotation invariance and grayscale invariance. According to [9], the LBP mask is set at 7 × 7 pixels, and the detection window is set at 16 × 16. Multiscale redundant contourlet decomposition can describe the local directional and structural properties of the fabric texture. According to [16], the RCT selects four directions and three levels.

Accuracy of the TILDA database

The fabric image of the C1 group in the TILDA database belongs to fabrics of the plain that form a tiny patch. Therefore, the patch size calculated by DMF cannot be directly used as the object of feature extraction. We have fixed the patch size to 28 × 28 pixels. In the transfer-learning stage, 645 defective patches and 430 defect-free patches are chosen for each category of fabric image. In the test phase, 10 original images are generated to patches. The program runs 10 times and reports average accuracy. The training accuracy of the TILDA database is 97.48%.

The test results are shown in Figure 9. Considering all the methods, the method of transfer learning is superior to the method of manually designed features, and our

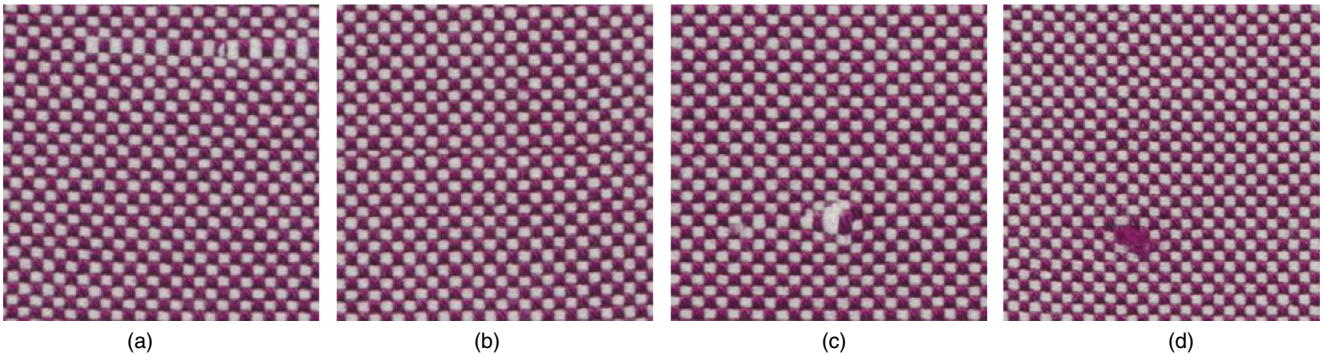

Figure 7 The representative samples for each class: (a) Carrying, (b) Thin Bar, (c) Knots, and (d) Fuzz Balls [Colour figure can be viewed at wileyonlinelibrary.com]

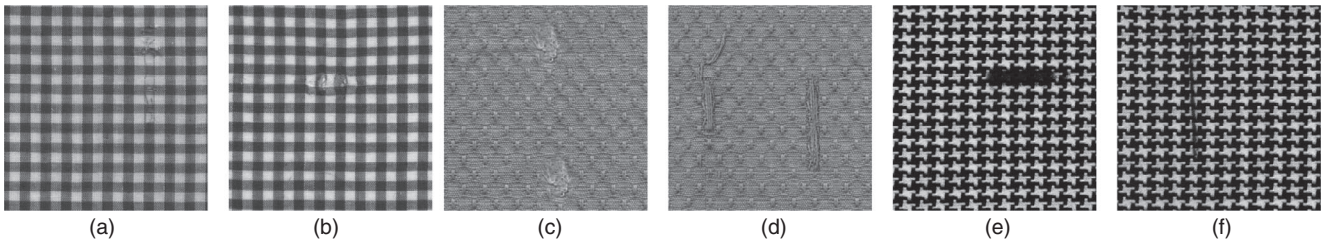

Figure 8 The typical pattern fabric samples: (a) BrokenEnd, (b) Hole, (c) Knot, (d) NettingMultiple, (e) Thick Bar, and (f) Think Bar

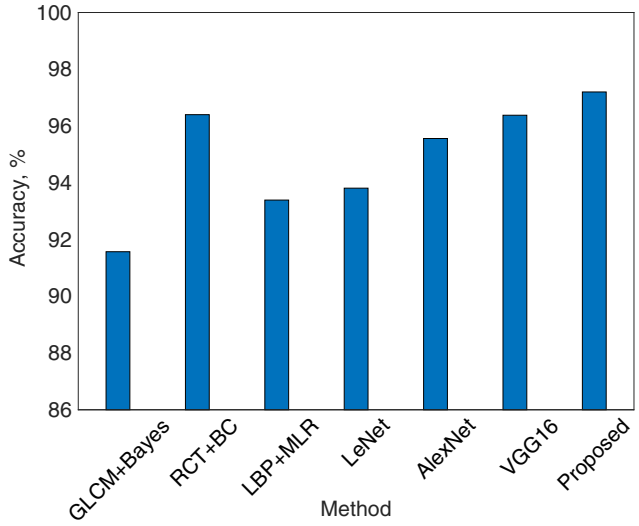

Figure 9 The test accuracy of seven methods on the TILDA database. The test accuracy of the proposed method is 97.20% [Colour figure can be viewed at wileyonlinelibrary.com]

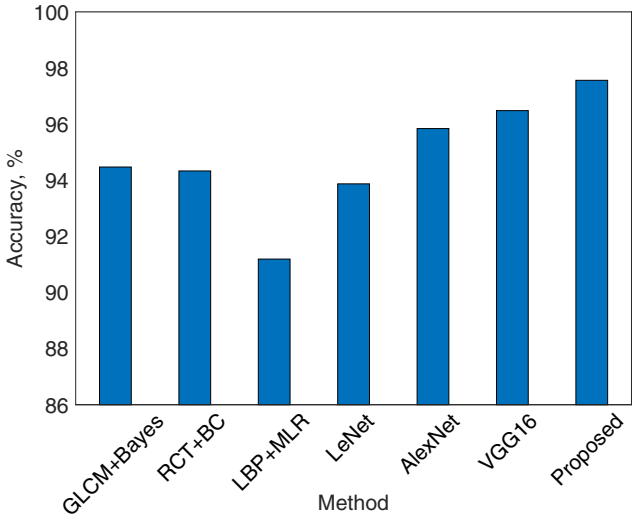

Figure 11 The test accuracy of seven methods on dark red fabric. The accuracy of the proposed method is 97.56% [Colour figure can be viewed at wileyonlinelibrary.com]

proposed method exhibits the best performance for this database.

Accuracy of dark red fabric

The dark red fabric image has a homogeneous texture on the image surface; the patch size calculated by DMF is 18 × 18 pixels, as shown by the black box in Figure 10a. If the 18 × 18 pixels patch size is used to divide the image directly, it is necessary to ensure that there is no rotation of the original image. Otherwise the information contained in the patch would change significantly compared with the previous patch, as shown in Figure 10b.

In a practical industrial environment, the images collected through the camera would have a small angle, so it is almost impossible to ensure that there is no rotation of the image. To improve the robustness of the algorithm, the patch size of the dark red fabric is set at 32 × 32 pixels, as shown by the white box in Figure 10a; the experimental results are as shown in Figure 11. The training accuracy of dark red fabric is 98.62%.

The test results show that our proposed method can achieve high accuracy with 97.56%. It outperforms traditional classification (GLCM, RCT and LBP) and other methods of transfer learning methods (LeNet, AlexNet and VGG).

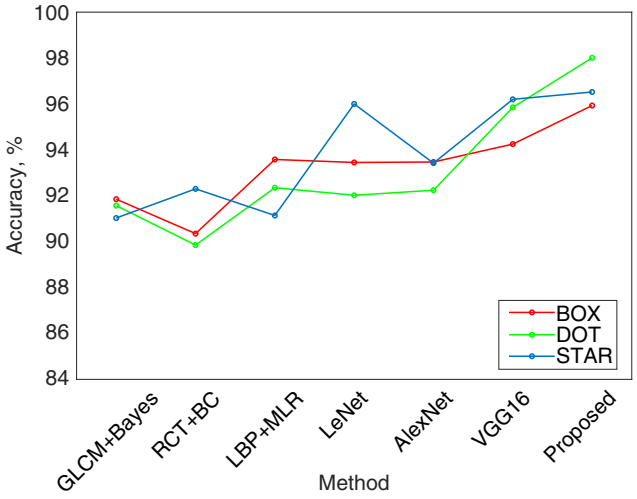

Figure 12 The test accuracy of seven methods on patterned fabric. For box-patterned fabric, the test accuracy of the proposed method is 95.93%. For dot-patterned fabric, the test accuracy of the proposed method is 98.01%. For star-patterned fabric, the test accuracy of the proposed method is 96.52% [Colour figure can be viewed at wileyonlinelibrary.com]

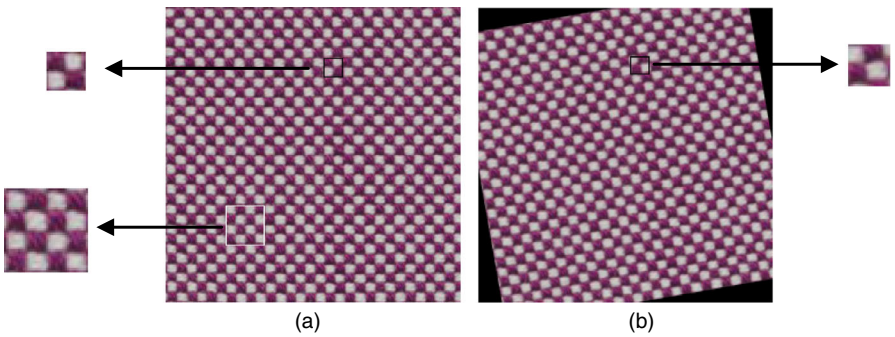

Figure 10 The patch size of dark red fabric. (a) The white box is a 32 × 32 patch, and the black box is an 18 × 18 patch; (b) 18 × 18 patch when the original image is rotated 5° [Colour figure can be viewed at wileyonlinelibrary.com]

Patterned fabric database

The patch sizes of the box-patterned fabric, the dot-patterned fabric and the star-patterned fabric calculated by DMF are 25 × 25, 27 × 37 and 17 × 21 pixels, respectively. The patch size of the star-patterned fabric is 48 × 48 pixels. The training accuracy of the box-patterned, dot-patterned and star-patterned fabric is 95.58, 98.40 and 96.89%, respectively.

From the experimental results shown in Figure 12, it can be seen that our proposed method exhibits the best performance for all three patterned texture fabrics. The red, green and blue lines indicate the accuracy of box-patterned, dot-patterned and star-patterned fabrics, respectively.

Evaluation of accuracy

Unlike most of the existing defect detection methods, our method uses the local patch as the input to the network, so therefore we should evaluate the performance of the model at the local level. The performance evaluation is described in [16].

In order to evaluate the performance of the model for defect location, Yapi *et al.* [16] proposed metrics at the local level, which are local precision (P_L), local recall (R_L) and local accuracy (ACC_L), defined as follows:

$$P_L = \frac{TP_L}{TP_L + FP_L} \times 100\% \tag{11}$$

$$R_L = \frac{TP_L}{TP_L + FN_L} \times 100\% \tag{12}$$

$$ACC_L = \frac{2 \times P_L \times R_L}{P_L + R_L} \times 100\% \tag{13}$$

where TP_L is true-positive, FP_L is false-positive, TN_L is true-negative, FN_L is false-negative, and ACC_L is the harmonic mean of P_L and R_L.

Parameter selection

Parameters for the model

Table 1 shows the different parameter settings in our model. In Table 1, Model-1 to Model-3 represent the settings of three models with different parameters, in which the convolution kernel size is 3 × 3 pixels, the activation function is ReLU, and a final layer uses a standard fully connected (FC) layer with one hidden layer. Model-4 has the same convolution

Table 1 Parameter settings of the deep convolutional neural network model and performance on MNIST

	Model-1	Model-2	Model-3 (proposed)	Model-4
Conv1	Kernel size = 3 Feature Map = 16 Stride = 1 Padding Size = 1 Activation = ReLU	Kernel size = 3 Feature Map = 16 Stride = 1 Padding Size = 1 Activation = ReLU	Kernel size = 3 Feature Map = 16 Stride = 1 Padding Size = 1 Activation = ReLU	Kernel size = 3 Feature Map = 16 Stride = 1 Padding Size = 1 Activation = ReLU
Sub2	Max Pooling = 2	Max Pooling = 2	Max Pooling = 2	Max Pooling = 2
Conv3	–	Kernel size = 3 Feature Map = 32 Stride = 1 Padding Size = 1 Activation = ReLU	Kernel size = 3 Feature Map = 32 Stride = 1 Padding Size = 1 Activation = ReLU	Kernel size = 3 Feature Map = 32 Stride = 1 Padding Size = 1 Activation = ReLU
Sub 4	–	Max Pooling = 2	Max Pooling = 2	Max Pooling = 2
Conv5	–	–	Kernel size = 3 Feature Map = 64 Stride = 1 Padding Size = 1 Activation = ReLU	Kernel size = 3 Feature Map = 64 Stride = 1 Padding Size = 1 Activation = ReLU
F6	1024 (2048)	1024 (2048)	1024 (**2048**)	2048
F7	10	10	10	1024 (2048)
F8	–	–	–	10
Max accuracy (%)	96.57 (95.85)	85.99 (91.05)	97.85 (**98.43**)	70.04 (76.13)

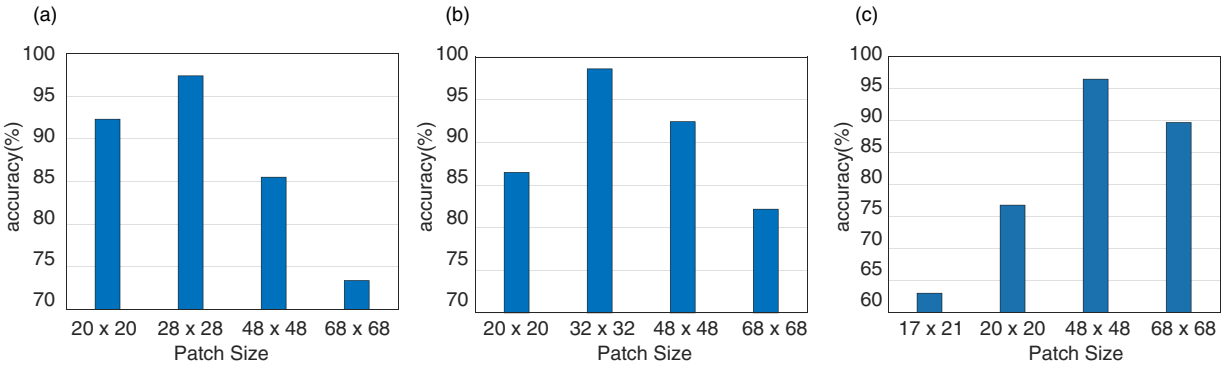

Figure 13 The accuracy of three kinds of fabrics with different patch sizes: (a) the fabric image of the C1 group; (b) dark red fabric; and (c) star-patterned fabric [Colour figure can be viewed at wileyonlinelibrary.com]

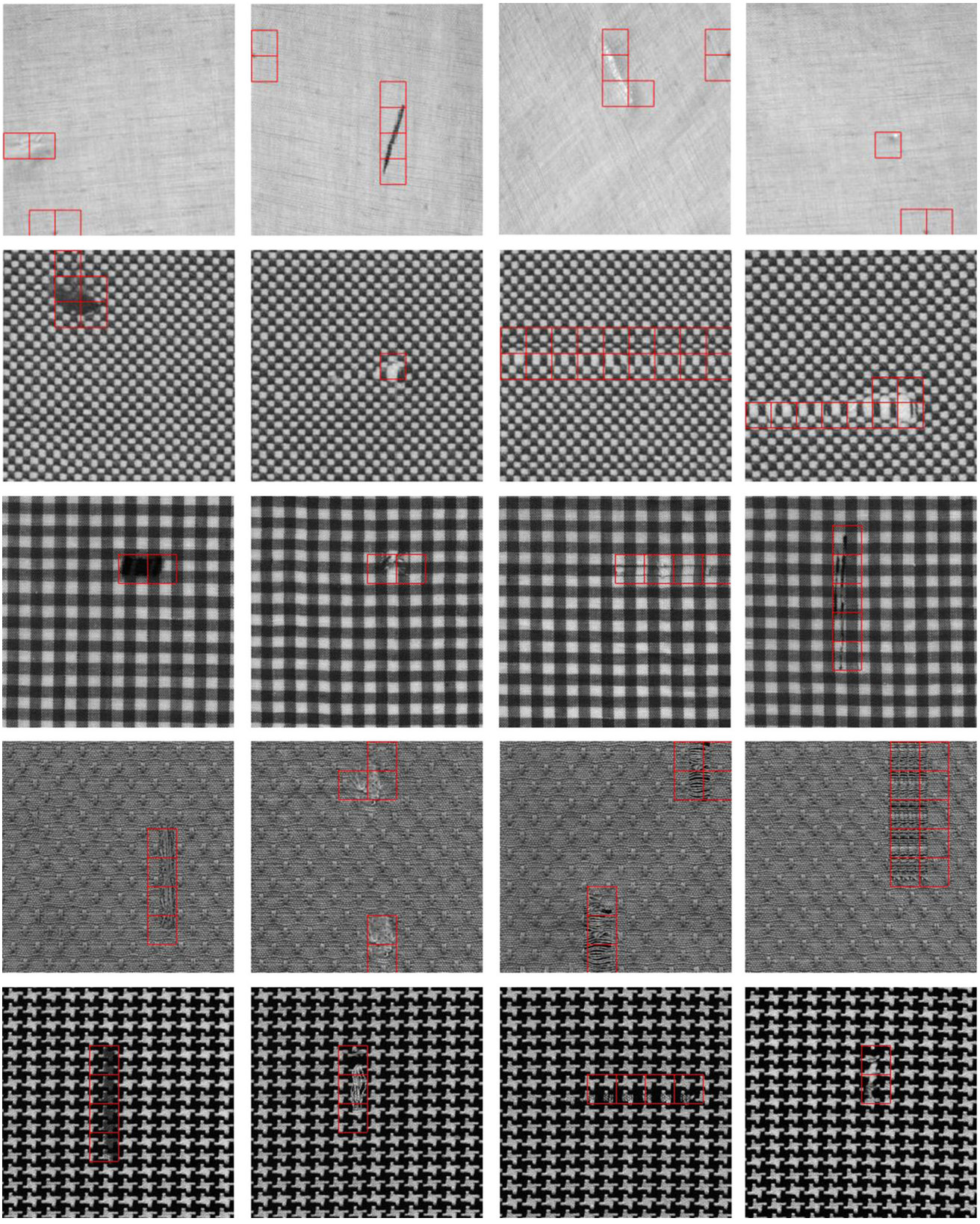

Figure 14 The fabric defect inspection results for the fabric image of the C1 group, dark red fabric, box-patterned fabric, dot-patterned fabric and star-patterned fabric, from the first to the fifth row [Colour figure can be viewed at wileyonlinelibrary.com]

layer as Model-3, while Model-4 has two hidden layers. The last row shows the performance of the four models on the MNIST dataset, and that Model-3 is superior to the other two models with 98.43% accuracy. Therefore, Model-3 is the structure proposed in our paper and its parameters are saved as the initialisation parameters of training fabric data, while preparing for the next stage of fabric data training.

Model-1 achieved an accuracy of 96.57% and 95.85% on the fully connected layers with 1024 neurons and 2048 neurons, respectively. However, Model-1 has a large generalisation error during the test phase, for which there is a serious overfitting phenomenon in the network. From Model-2 to Model-3, the accuracy increases with the number of convolution layers and the number of neurons in the FC layer. The

Table 2 The test accuracy of seven methods on three fabric databases

	TILDA	YIDA	BOX	DOT	STAR	Avg-ACC
GLCM						
P_L (%)	90.07	93.05	93.54	91.35	92.49	92.10
R_L (%)	93.13	95.94	90.22	91.79	89.61	
ACC_L (%)	91.57	94.47	91.85	91.57	91.03	
RCT						
P_L (%)	97.50	96.69	89.52	92.58	88.46	92.65
R_L (%)	95.32	92.08	91.20	89.85	96.49	
ACC_L (%)	96.40	94.33	90.35	89.85	92.30	
LBP						
P_L (%)	95.39	89.77	94.06	93.45	90.66	92.33
R_L (%)	91.48	92.66	93.10	91.27	91.62	
ACC_L (%)	93.39	91.19	93.58	92.35	91.14	
LeNet-5						
P_L (%)	89.53	97.34	93.89	94.67	93.54	93.83
R_L (%)	98.52	90.63	93.01	89.52	98.60	
ACC_L (%)	93.81	93.87	93.45	92.02	96.00	
AlexNet						
P_L (%)	94.37	96.27	90.74	93.87	91.86	94.10
R_L (%)	96.40	95.41	96.36	90.67	95.03	
ACC_L (%)	95.56	95.84	93.47	92.24	93.42	
VGG16						
P_L (%)	97.15	97.75	98.34	96.86	95.63	96.03
R_L (%)	95.62	95.24	90.48	94.86	96.78	
ACC_L (%)	96.38	96.48	94.25	95.85	96.20	
Proposed						
P_L (%)	96.04	96.19	96.86	96.42	94.85	97.31
R_L (%)	98.39	98.97	95.02	99.65	98.25	
ACC_L (%)	97.20	97.56	95.93	98.01	96.52	

Table 3 Trainable parameters in the deep convolutional neural network

	LeNet-5	AlexNet	VGG16	Proposed
Parameters	61 772	56 876 418	134 281 029	3244 800

maximum accuracy is 98.43% when there are 2048 neurons in the FC layer (As shown in bold in Table 1). At the same time, Model-4 adds one layer of FC layer based on Model-3, and the accuracy rate is only 76.13%, which indicates that the MNIST dataset is insufficient to train the parameters in Model-4.

As reported in Table 1, if the model complexity is too low for the given dataset, there is a high probability that the overfitting phenomenon will occur. Conversely, the underfitting phenomenon may occur. The number of neurons in the full connection layer is not as large as possible. Generally, the value is 2^n in the condition, which is less than the output of the previous layer.

Table 5 The accuracy of using pretraining and not using pretraining

	No pretraining		Pretraining	
	Training accuracy (%)	Test accuracy (%)	Training accuracy (%)	Test accuracy (%)
TILDA	86.35	75.17	97.48	97.20
Dark red fabric	89.34	68.60	98.62	97.56
Box-patterned fabric	79.62	60.45	95.58	95.93
Dot-patterned fabric	82.38	50.57	98.40	98.01
Star-patterned fabric	78.48	71.43	96.89	96.52

Patch size selection

For regular patterned fabrics, the texture cycle of the fabric surface can be automatically and accurately calculated by DMF, which automatically determines the patch size. However, for plain weave and twill weave fabrics with the smaller period (e.g. the fabric image of C1 group, dark red fabric, and star-patterned fabric), if we directly use the patch size calculated by DMF as the period to divide the image, the texture information contained in the patch will be insufficient. It will ultimately affect the accuracy of the detection result. Given the three types of fabric mentioned above, the patch size is set by hand to train the model, and the accuracy of multiple different patch size models is tested. The experimental results are shown in Figure 13.

The selection of the patch size depends on the size of defect. The results show that if the value of the patch size is too small, i.e. 20 × 20 pixels and 17 × 21 pixels, which might not include enough information for the defect, the network cannot learn the proper characteristics. If the value of the patch size is too large, i.e. 48 × 48 pixels and 68 × 68 pixels, the patch might contain too much redundant information, which may reduce the performance of the classification and location. The principle of selecting patch size is to ensure that each patch includes at least one complete texture period. Therefore, choosing the appropriate patch size is a crucial step in the proposed method.

Results and Discussion

We have compared our method with other methods including GLCM, RCT, LBP, LeNet, AlexNet and VGG16 in the same fabric database. The GLCM method often needs to manually set the threshold between the template image and the image, and it is easy to fall into the dimension disaster in the process of computing the Euclidean distance. The statistical characteristics of the fabric image are obtained by fitting the coefficients of RCT using a finite mixture of a generalised Gaussian model, which has invariance to fabric translation and scale changes. The

Table 4 Prediction time for seven methods and training time of 3000 iterations

	GLCM	RCT	LBP	LeNet-5	AlexNet	VGG16	Proposed
Prediction on Central Processing Unit (CPU) (ms)	184	2090	465	23	287	3331	89
Prediction on Graphics Processing Unit (GPU) (ms)	–	–	–	3	42.6	385	25
Training time (min)	–	–	–	49	115	247	35

disadvantage of this method is that the fitting process is lengthy, which leads to the slow speed of the algorithm and poor real-time performance. Similarly, the LBP operator can also describe the texture information of the fabric, but the number of the sampling points has a great influence on the extraction and recognition of texture features.

AlexNet uses five convolutional layers and a large number of feature maps to extract features of the input image and map the features layer by layer to high dimensions, making the features easier to distinguish. The average accuracy on the five fabric images is 94.10%. The disadvantage of using AlexNet is that the input size of the network is fixed to 227×227 pixels. It is necessary to upsample the patch (size not exceeding 50×50) to 227×227 pixels in the preprocessing stage. This process would lose a large amount of original information in the image and affect the extraction and classification of the features. VGG16 divides the network into five groups (simulating the five layers of AlexNet), using only 3×3 convolution kernels, and combining them as a convolution sequence for processing. The network is deeper and the number of channels is larger. The innovation is the usage of multiple 3×3 convolutions to simulate larger receptive fields, which can make the convolution layer more nonlinear with fewer parameters. These ideas are also extensively applied to subsequent network architectures such as Inception [51] and ResNet [52]. However, its disadvantages are the same as using AlexNet.

The performance of each algorithm is evaluated using the P_L, R_L and ACC_L in [16], and the location of defects is marked on each test image (the result is shown in Figure 14). The average accuracy of each method is shown in Table 2. It is shown that the method of manually designed features (e.g. GLCM, RCT and LBP) needs to change the parameters in the feature extractor for different types of fabrics. The range of ACC is between 89–93%. The method of transfer learning is generally superior to traditional detection methods, and the range of ACC is between 93–98%. Among all the methods, the proposed method has the highest average accuracy.

The complexity of our network model is between LeNet-5 and AlexNet. Compared with LeNet-5, the number of convolution kernels per layer is increased, so that the model can learn more features. The test result is 1.3–5.3%, which is higher than that of LeNet-5. Compared with AlexNet and VGG16, our method can achieve the highest average accuracy, and the parameters of our model are less and our model has better real-time capability. Table 3 shows the parameters of the deep CNN. Table 4 shows the prediction time for each algorithm during the testing phase and the time required for 3000 iterations. Table 5 shows the results of both using and not using pretraining. In the case of using pretraining, there is a larger gap between the training results and the test results, and the highest result for training accuracy is only 86.35%, indicating an obvious overfitting phenomenon. On the contrary, we used the method of pretraining, augmentation data and dropout to combat overfitting, and achieved higher training accuracy and test accuracy. The last line in Table 4 is the time required for the four neural networks to iterate 3000 times. Compared with the other three models, our model spends less time in the training process.

In the manual design feature method, the GLCM algorithm has the shortest computational time. LBP algorithms traverse the whole image through a sliding window during feature extraction, so it takes a longer computation time. The 80% computation time of RCT's method is used to fit a mixture of a generalised Gaussian model.

In the method of transfer learning, the LeNet-5 model has fewer parameters but also significantly lower accuracy than our model. AlexNet and VGG contain multiple convolutional layers and feature maps, so there are a large number of parameters in the model. The number of parameters of the proposed model is only 2% of the number of the VGG parameter. It can satisfy the requirements of high accuracy and real-time performance in industrial inspection.

Conclusions

We proposed a new fabric defect detection algorithm which can deal with various types of fabrics. Our method does not directly use the original image as input. Instead, we divide the fabric image into multiple patches along the inherent period of the fabric surface, which is used as an operation object to train deep CNN. Our algorithm achieved an average accuracy of 97.31% for results on three datasets, which can achieve accurate detection of common defects in yarn-dyed fabric, such as Carrying, Thin Bar, Scratch, Knots, BrokenEnd, Stain, and Holes. Compared with traditional shallow learning approaches, the experimental results demonstrate that our proposed method can effectively learn defect features by adaptively adjusting the parameters. In addition, our method can improve efficiency, shortening the time of measurement, and obtaining an accurate defect image.

In the future, we will focus on two directions of research. One direction involves the defect segmentation. The other direction is to automate the period of the texture process using deep learning methods.

Acknowledgements

The authors gratefully thank the Scientific Research Program Funded by Natural Science Foundation of China (Grant No. 61301276), the Key Research and Development plan of Shaanxi province (Grant No. 2017GY-003), Youth Talent Promotion Project of Shaanxi Association for Science and Technology (Grant No. 20180115), and the Scientific Research Program Funded by Shaanxi Provincial Education Department (Grant No. 18JK0338).

References

1. H Y T Ngan, G K H Pang and N H Yung, *Image Vis. Comput.*, **29** (2011) 442.
2. A Kumar, *IEEE Trans. Ind. Electron.*, **55** (2008) 348.
3. M Bennamoun and A Bodnarova, *Proc. IEEE Conf. SMC.*, San Diego, CA, USA (1998) 4340.
4. C S Cho, B M Chung and M J Park, *IEEE Trans. Ind. Electron.*, **52** (2005) 1073.
5. X Z Yang, G K H Pang and N H C Yung, *Opt. Eng.*, **41** (2002) 3116.
6. C Kwak, J A Ventura and K Tofang-Sazi, *Intell Data Anal.*, **5** (2001) 355.
7. R T Chin and C A Harlow, *IEEE Trans. Pattern Anal. Mach. Intell.*, **4** (1982) 557.
8. E J Wood, *Text. Res. J.*, **60** (1990) 212.

9. F Tajeripour and E Kabir, *Proc. of the Int. Conf. on Comput. Intel. and Multimed. Appl.*, Sivakasi, Tamil Nadu, India (2007) 261.
10. C H Chan and G Pang, *IEEE Trans. Ind. Appl.*, **36** (2000) 1267.
11. D M Tsai and T Y Huang, *Image Vis. Comput.*, **21** (2003) 307.
12. D M Tsai and C H Chiang, *Image Vis. Comput.*, **21** (2003) 413.
13. J Jing, A Dong and P Li, *Opt. Eng.*, **56** (2017) 1.
14. L H Hoffer, F Francini, B Tiribilli and G Longobardi, *Opt. Eng.*, **35** (1996) 3183.
15. D Zhu, R Pan, W Gao and J Zhang, *Autex Res. J.*, **15** (2015) 226.
16. D Yapi, M Allili and N Baaziz, *IEEE Trans. Autom. Sci. Eng.*, **99** (2017) 1.
17. G Oh, S Lee and S Yong Shin, *Pattern Recognit. Lett.*, **20** (1999) 191.
18. Y LeCun, B E Boser, J S Denker, D Henderson, R E Howard, W E Hubbard and L D Jackel, *NIPS.*, Denver, CO, USA (1989) 396.
19. H Y T Ngan, G K H Pang and N H C Yung, *Pattern Recogn.*, **41** (2008) 1878.
20. A Latif-Amet, A Ertüzün and A Erçil, *Image Vis. Comput.*, **18** (2000) 543.
21. H Sari-Sarraf and J S Goddard, *IEEE Trans. Ind. Applicat.*, **35** (1999) 1252.
22. X Yang, G Pang and N Yung, *Pattern Recogn.*, **37** (2004) 889.
23. H Y T Ngan, G K H Pang, S P Yung and M K Ng, *Pattern Recogn.*, **38** (2005) 559.
24. C Guo, Q Ma and L Zhang, *CVPR.*, Anchorage, AK, USA (2008) 1063.
25. J G Campbell, A A Hasim and F D Murtagh, *ISSC* (Ireland: Univ. Ulster, 1997) 97.
26. F S Cohen, Z Fau and S Attali, *IEEE Trans. Pattern Anal. Mach. Intell.*, **13** (1991) 803.
27. M S Allili, N Baaziz and M Mejri, *IEEE Trans. Multimedia*, **16** (2014) 772.
28. J Long, E Shelhamer and T Darrell, *CVPR* (Boston, MA, USA: Hynes Convention Center, 2015) 1.
29. Y LeCun, Y Bengio and G Hinton, *Nature*, **521** (2015) 436.
30. S Faghih-Roohi, S Hajizadeh, A Núñez, R Babuska and B D Schutter, *Proc. Int. Joint Conf. Neural Netw.*, Vancouver, BC, Canada (2016) 2584.
31. X Bian, S N Lim and N Zhou, *Proc. WACV*, Lake Placid, NY, USA (2016) 1.
32. D Cireşan, U Meier, J Masci and J Schmidhuber, *Neural Netw.*, **32** (2012) 333.
33. Y Lou, G Fu, Z Jiang, A Men and Y Zhou, *IEEE GlobalSIP*, Montreal, QC, Canada (2017) 1280.
34. P Sermanet, K Kavukcuoglu, S Chintala and Y Lecun, *CVPR* (Portland, OR, USA: Oregon Convention Center, 2013) 3626.
35. R Vaillant, C Monricq and Y Le Cun, *IEE Proc. Vis. Image Signal Process.*, **141** (1994) 245.
36. C Garcia and M Delakis, *IEEE Trans. Pattern Anal. Mach. Intell.*, **26** (2004) 1408.
37. Y Taigman, M Yang, M Ranzato and L Wolf, *CVPR* (Columbus, OH, USA: Greater Columbus Convention Center, 2014) 1701.
38. J Jing, X Fan and P Li, *J. Text. Res.*, **38** (2017) 68.
39. R Ren, T Hung and K C Tan, *IEEE Trans. Cybern.*, **48** (2017) 929.
40. P F Felzenszwalb and D P Huttenlocher, *Int. J. Comput. Vis.*, **59** (2004) 167.
41. J Jing, P Yang and P Li, *J. Text. Res.*, **36** (2015) 98.
42. A Santoro, S Bartunov, M Botvinick, D Wierstra and T Lillicrap, *International Conference on Machine Learning*, New York City, NY, USA (2016) 1842.
43. J Snell, K Swersky and R Zemel, *NIPS* (Long Beach, CA, USA: Long Beach Convention Center, 2017) 30.
44. O Vinyals, C Blundell, T Lillicrap, K Kavukcuoglu and D Wierstra, *NIPS* (Barcelona Spain: Centre Convencions Internacional Barcelona, 2016) 29.
45. H C Shin, H R Roth, M Gao, L Lu, Z Xu, I Nogues, J Yao, D Mollura and R M Summers, *IEEE Trans. Med. Imaging*, **35** (2016) 1285.
46. J Lu, V Behbood, P Hao, H Zuo, S Xue and G Zhang, *Knowl.-Based Syst.*, **80** (2015) 14.
47. Y Lecun, L Bottou, Y Bengio and P Haffner, *Proc. IEEE*, **86** (1998) 2278.
48. J Deng , W Dong, R Socher, L Li, K Li and F Li, *CVPR*, Miami, FL, USA (2009) 248.
49. A Krizhevsky, I Sutskever and G Hinton, *NIPS*, Lake Tahoe, NV, USA (2012) 1095.
50. K Simonyan and A Zisserman, *ICLR*, San Diego, CA, USA (2015).
51. C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke and A Rabinovich, *CVPR* (Boston, MA, USA: Hynes Convention Center, 2015) 1.
52. K He, X Zhang, S Ren and J Sun, *CVPR* (Boston, MA, USA: Hynes Convention Center, 2015) 770.

