Real-time fabric defect detection based on multi-scale convolutional neural network

Shuxuan Zhao¹, Li Yin¹, Jie Zhang¹, Junliang Wang¹, Ray Zhong²

¹College of Mechanical Engineering, Donghua University, Shanghai 201620, People’s Republic of China
²Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong 999077, People’s Republic of China

Abstract: Fabric defect detection plays an important role in ensuring quality control in the textile manufacturing industry. This study introduces a fabric defect detection method based on a multi-scale convolutional neural network (MSCNN) to improve accuracy and time efficiency. For detection accuracy, the MSCNN is constructed to obtain different scales of feature maps, which enhance the representation of tiny scale fabric defects. A faster defect locating method is designed with pre-known size information obtained by clustering analysis to reduce the computation time. An experiment is carried out for illustrating that the accuracy of MSCNN for each defect reaches over 92%, and the frames per second (FPS) is more than 29. Further analysis results demonstrate that the proposed MSCNN can accurately detect the fabric defects with a tiny scale, and the speed of detection can reach 30 m/min to satisfy the industrial requirements.

1 Introduction

Fabric defect detection is significant for textile quality control [1]. Traditionally, the defects are examined by skilled workers via visual sensation, as shown in Fig. 1. The manual detection method has disadvantages as a high error rate for human fatigue and low-speed execution as skilled workers can detect only 15–20 m/min. Therefore, it’s necessary to develop an automatic detection method to enhance productive efficiency [2].

In recent years, with the rapid development of digital image processing techniques and machine vision, lots of automatic detection methods are proposed to replace manual detection. The fabric defect detection methods based on machine vision are mainly divided into two groups: non-motif-based and motif-based groups. The motif-based methods require a defect-free ground truth for comparison of the motifs, but there are plenty of different fabrics with complicated background features, and even the same type of fabric has a difference because of the exchanging workshop environment, such as the noise and illumination. It is hard for motif-based methods to obtain the robust defect-free ground truth for detection [3]. Therefore, most works focused on the non-motif-based group, which can be further subdivided into five categories: statistical, spectral, model-based, learning-based, and structural [4], as shown in Fig. 2. The statistical methods [5–7] distinguish defective and defect-free regions according to the analysis of their different statistical feature, similarity, and regularity. This method is not efficient in detecting low-contrast fabrics and patterned fabrics. The spectral methods [8, 9], widely used in the study of fabric defects, highlight the differences between defective and defect-free regions in the frequency domain, which can effectively realise the detection of fabrics with a simple background. Typical spectral methods include Fourier transform, wavelet transform [10], and Gabor transform [11]. Although these methods are widely studied, they cannot effectively detect patterned fabrics. Also, the real-time requirements are difficult to meet because they are high computationally demanding. The model-based and structural methods [12] are relatively less, possibly because they are not robust enough and highly data-dependent. In this paper, we will mainly concentrate on the learning-based method.

With the rapid development of artificial intelligent technology, deep learning method has achieved brilliant performance in image classification and detection. Numerous learning-based methods [13–18] have been proposed to solve the problems of fabric defect detection because of its greater extraction ability of image features. Jing et al. [19] proposed a learning-based method with a deep convolutional neural network (CNN) to extract the features of fabric, and used the mean shift algorithm to segment the defects, realising the detection of Yarn-dyed fabric. However, this method needs two steps to realise the detection of fabric, cannot realise the end-to-end detection of detection. Wei et al. [20] used a learning-based method to construct faster region-CNN (R-CNN) to classify and search defects region, acquiring high accuracy end-to-end detection of fabrics with complicated texture and small defects. While this method requires lots of computational capacity, meaning to be time consuming, which cannot meet the real-time...
The CNN is used to extract features from the image matrix, and obtain the feature map with the size of all types defects, and then using these defects bounding boxes with pre-known size information, which are used to replace existing time-consuming locating methods, so as to reduce the computation in location and improve the speed of detection.

The rest of this paper is organised as follows: Section 2 presents the MSCNN framework and the model training process. Section 3 reports the experiments and discussion. Section 4 shows the conclusion and raises several future issues.

2 Proposed fabric defects detection method

2.1 Framework of MSCNN model

Aiming at existing fabric defects detection methods have low accuracy for detecting tiny-scale defects, we take advantage of the strong ability of pyramid features [27] for extracting multi-scale features, apply the pyramid features a method in neural network and propose the MSCNN to improve the detection accuracy for tiny scale defects. The MSCNN is constructed using feature fusion based on the VGG16 network [28].

To satisfy the requirement of fabric detection, we take advantage of that the same type of defects always have similar shapes and sizes, using a clustering algorithm to get the common features of all types defects, and then using these defects bounding boxes with pre-knowledge to replace the region generation method in defects locating, so as to reduce the computation required in the defects location process and to improve the speed of detection. The framework of the proposed MSCNN is shown in Fig. 4.

2.1.1 Fabric image feature extraction: In the progress of detection: the fabric images captured by the line-scan charge-coupled device (CCD) camera are used as input, and standardised as the matrix $\mathcal{X} = [x_1, \ldots, x_l]$. The CNN is used to extract features from the image matrix, and obtain the feature map with the pixel of 13*13 and 26*26, respectively. The feature map is input in the detection module, and the output is location and classification information of fabric defects detection.

At the front end of the CNN, the features of fabric images are extracted using convolution and pooling. The 3*3 size and 1 step small convolution kernels are used in the convolutional layer to enhance the ability to extract the local features of the fabric images; the convolution computation formula is shown as

\[ \mathcal{F} = \sum_{i} \sum_{j} x_{i+j} \cdot \mathcal{W}_{i,j} \]

where $\mathcal{F}$ is the output feature map, $x_{i+j}$ is the input pixel, and $\mathcal{W}_{i,j}$ is the convolution kernel.

requirement of detection in fabric production. Liu et al. [21] proposed a single shot multibox detector (SSD) method for fabric defect detection; the detector is designed directly at the end of the CNN, so that the speed of detection has been improved. However, this method failed to locate some tiny defects accurately. Although the detection algorithm based on deep learning has better performance in accuracy and more robust than other detection methods. There are still two main challenges in the defect detection: the fabric images captured by the line-scan charge-coupled device (CCD) camera are used as input, and standardised as the matrix $\mathcal{X} = [x_1, \ldots, x_l]$. The CNN is used to extract features from the image matrix, and obtain the feature map with the pixel of 13*13 and 26*26, respectively. The feature map is input in the detection module, and the output is location and classification information of fabric defects detection.

First, the defects with tiny scales are difficult to detect, and the existing fabric defect detection methods do not design a network aiming at this character. As shown in Fig. 3, the hole and stain defects, as the common defects, have regular shapes and scales, which are relatively easy to detect. However, the end out defects, double flat and broken course defects are tiny in scale with irregular shapes. These defects are long and narrow in shape, only a few pixels, accounting for less than 1/100 of the whole image's pixels in one direction. Compared to the defects with regular shapes, these defects with irregular shapes and tiny scales are more difficult to extract effective defects features from the fabric images detected, which improves the complexity of detection.

Second, the fabric production has a high requirement for real-time defect detection; the movement speed of fabrics in production is often higher than 15 m/min. While the existing deep-learning detection methods do not consider the similarity shapes and sizes of the same types of fabric defects when designing the detector, which always causes time-consuming of detection, it can hardly satisfy the real-time demand of fabric detection, and will lead to a decrease in productivity. That means they cannot be applied in practice.

In this paper, a deep-learning-based detection method, multi-scale CNN (MSCNN), is proposed to realise the detection for five types of defects in linen fabric and patterned fabric. Compared with other detection methods, there are two main contributions:

(i) Targeting at existing fabric defects detection methods have low accuracy in detecting tiny-scale defects, applies the pyramid features a method in a neural network to enhance the feature representation and detection accuracy of tiny scale fabric defects.

(ii) For real-time detection, taking advantage of the similarity of the same type of defects, using the K-means clustering analysis to obtain defects bounding boxes with pre-known size information, which are used to replace existing time-consuming locating methods, so as to reduce the computation in location and improve the speed of detection.
2.1.2 Defects location and classification: In the fabric production process, the two main factors determining the shape and size of defects are fabric type and production equipment. Although the shape of defects varies from this fabric to another, there is no significant difference in shape and size between the same defects. In order to reduce the computation of location and to improve the speed of detection, the locating method is designed with pre-known size information obtained by clustering analysis [29, 30] to reduce the computation time of defect region generation as shown in Fig. 5.

The defects bounding boxes with pre-known size information is obtained K-means clustering algorithm; the clustering algorithm is shown as

\[ J = \min \left( \sum_{i=1}^{k} \sum_{j=1}^{N} \| x_i - u_k \| ^2 \right) \]

where \( x_i \) are the coordinates \( \{w_{box}, h_{box}\} \) of defects bounding boxes in datasets, \( u_k \) are the coordinates \( \{w_{box}, h_{box}\} \) of defects bounding boxes with the pre-known size we want to obtain, and \( u_k \) are randomly assigned to an initial value at first, \( k \) is the number of defects bounding box types. In this study, we set \( k = 4 \) corresponding to four types of defects. By iterating and minimising the \( J \), the defects bounding boxes \( x_i \) are divided into four categories according to their size and aspect ratio.

After obtaining the categories of defects, The intersection over union (IOU ) between each type of defect bounding boxes with pre-known size and corresponding defect bounding box was calculated, respectively. The IOU represents the overlap ratio, and the higher the IOU is, the higher the overlap degree will be. When the intersection ratio value is one, it means that the defects bounding boxes with pre-known defects size and corresponding defect bounding box completely coincide. The formula of IOU is shown as

\[ \text{IOU}_i = \frac{\text{Box}(w_{box}, h_{box}) \cap \text{Box}(w_{box}, h_{box})}{\text{Box}(w_{box}, h_{box}) \cup \text{Box}(w_{box}, h_{box})} \]

(4)

\[ \text{Box}(w_{box}, h_{box}) \cap \text{Box}(w_{box}, h_{box}) = \min (w_{box}, w_{box}) \times \min (h_{box}, h_{box}) \]

(5)

(see (6)) where \( \text{Box}(w_{box}, h_{box}) \) represent the defects bounding boxes with the pre-known defects size corresponding to \( (w_{box}, h_{box}) \) in length and width.

Although set \( D \) to be the optimisation parameters of IOU

\[ D = \frac{1}{N} \sum_{i=0}^{N} (1 - \text{IOU}_i)^2 \]

(7)

The gradient descent algorithm is used to optimise the parameters \( D \). After many iterations, the maximum size parameters of the...
defects bounding boxes with pre-known size for each type of defects are obtained; the gradient descent formula is shown as

\[ w_{box}^{t+1} = w_{box}^t - \alpha \frac{\partial D}{\partial w_{box}^t} \]  

\[ h_{box}^{t+1} = h_{box}^t - \alpha \frac{\partial D}{\partial h_{box}^t} \]  

After obtaining the size parameters of the defects bounding boxes with pre-known size, the feature map of the last convolution is input to detection. Although CNN's can extract fabric image features efficiently, it is hard to locate fabric defects. To locate the defects, it is necessary to match the defects bounding boxes with the feature map. Firstly, the scaling ratios \( p_w \) and \( p_h \) are obtained according to the matrix column-to-column ratio between the feature and the defects bounding boxes. Secondly, after obtaining the scaling ratios, the size parameters \( w_{box}^k, h_{box}^k \) of the defects bounding boxes of four categories are scaled in turn, and the size parameters \( (b_w, b_h) \) of the defects bounding boxes corresponding to the feature map are obtained, which are shown as

\[ b_w = p_w e^{h_i} \]  

\[ b_h = p_h e^{h_i} \]  

Thirdly, the feature map is divided into \( N*N \) cells according to the columns and rows of the feature map, the centre of each cell is obtained as the anchor coordinates \( (t_x, t_y) \). Each anchor coordinate is matched with the corresponding four types of defects bounding boxes, and only generate a total number of \( N*4 \) defects bounding boxes.

Finally, the classification probability \( P(i) \) of the defects and the confidence of defects bounding boxes are computed, which are shown as

\[ P(i) = \frac{e^h}{\sum e^h} \]  

\[ \text{Confidence} = P(i)*\text{IOU}_i \]  

where \( h \) represents the value of the convolution kernels corresponding to the last convolutional layer.

The confidence threshold value of the defects bounding box is set according to the detection requirements, and the threshold segmentation is used to remove the defects bounding boxes with a confidence lower than the threshold, so as to output the final fabric defect detection result, which is shown as

\[ Y = \begin{cases} 0 & \text{if confidence} \leq T \\ 1 & \text{otherwise} \end{cases} \]  

| Layer | Size | Filters | Input | Output |
|-------|------|---------|-------|--------|
| 1     | Conv | 3*3/1  | 16    | 416*416*3 |
| 2     | MaxP | 2*2/2  | —     | 416*416*16 |
| 3     | Conv | 3*3/1  | 32    | 208*208*16 |
| 4     | MaxP | 2*2/2  | —     | 208*208*32 |
| 5     | Conv | 3*3/1  | 64    | 104*104*32 |
| 6     | MaxP | 2*2/2  | —     | 104*104*64 |
| 7     | Conv | 3*3/1  | 128   | 52*52*64  |
| 8     | MaxP | 2*2/2  | —     | 52*52*128 |
| 9     | Conv | 3*3/1  | 256   | 26*26*128 |
| 10    | MaxP | 2*2/2  | —     | 26*26*256 |
| 11    | Conv | 3*3/1  | 512   | 13*13*256 |
| 12    | MaxP | 2*2/2  | —     | 13*13*512 |
| 13    | Conv | 3*3/1  | 256   | 13*13*256 |
| 14    | Conv | 3*3/1  | 256   | 13*13*256 |
| 15    | Conv | 1*1/1  | 27    | 13*13*256 |

Fig. 5 Fabric defect detection with pre-known defects size bounding boxes

where
\[ \text{Box}(w_{box}^i, h_{box}^i) \cup \text{Box}(w_{box}^k, h_{box}^k) = w_{box}^i \ast h_{box}^i + w_{box}^k \ast h_{box}^k \]
\[ + \min(w_{box}^i, w_{box}^k) \ast \min(h_{box}^i, h_{box}^k) \]
where the $T$ represents the threshold, and the $Y$ represents the final output defects bounding box in the detection results.

### 2.2 Model training

Image brightness changes and image noise caused by light changes and floating dust often appear in fabric image acquisition, which will influence the detection accuracy. The illumination normalisation [31] and noise corruption are processed to simulate the images of fabric in different workshop environments before the model training, so as to enhance the robustness for the detection model to environmental changes.

The model training takes the fabric images and defects bounding boxes with pre-known defects size as input, and improves the detection accuracy by minimising the loss function, the procedures of model training are as follows:

**Input:** Dataset $\mathcal{X}$, defects bounding boxes with pre-known defects size $u_k = \{w_{box}^k, h_{box}^k\}$, number of epochs.

**Process**

1. **Step 1:** Randomly set network parameters $W^{(b)}, b^{(b)}$, and centre of anchor coordinates $x_i, y_i$.
2. **Step 2:** Iterative update parameters for $i = 1$ to $m$
   1. **Calculate the partial derivatives:**
      - $\nabla W^{(b)} = \left(\frac{\partial \text{loss}}{\partial W^{(b)}}\right)$,
      - $\nabla b^{(b)} = \left(\frac{\partial \text{loss}}{\partial b^{(b)}}\right)$,
      - $\nabla x_i = \left(\frac{\partial \text{loss}}{\partial x_i}\right)$,
      - $\nabla y_i = \left(\frac{\partial \text{loss}}{\partial y_i}\right)$,
      - $\nabla w_{box} = \left(\frac{\partial \text{loss}}{\partial w_{box}}\right)$,
      - $\nabla h_{box} = \left(\frac{\partial \text{loss}}{\partial h_{box}}\right)$.
   2. **Update these parameters:**
      - $W^{(b)} = W^{(b)} + \alpha \nabla W^{(b)}$, $b^{(b)} = b^{(b)} + \alpha \nabla b^{(b)}$,
      - $x_i = x_i + \alpha \nabla x_i$, $y_i = y_i + \alpha \nabla y_i$,
      - $w_{box} = w_{box} + \alpha \nabla w_{box}$, $h_{box} = h_{box} + \alpha \nabla h_{box}$.
3. **Finish the iteration epochs**

**Output:** detection model

Where $\alpha$ is the learning rate, which is a variable constant. The epoch is the number of images updated each time. Loss consists of four parts: central coordinate loss $XY_{\text{loss}}$, bounding boxes loss $WH_{\text{loss}}$, classification loss $\text{Class}_{\text{loss}}$, and confidence loss $\text{Confidence}_{\text{loss}}$, which are shown as:

$$XY_{\text{loss}} = \lambda_{\text{coord}} \sum_{i=0}^{N} \sum_{j=0}^{B} e_{ij}^c [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]$$  \hspace{1cm} (15)

$$WH_{\text{loss}} = \lambda_{\text{coord}} \sum_{i=0}^{N} \sum_{j=0}^{B} \left(\sqrt{w_{box}^i - w_{box}^j} + \sqrt{h_{box}^i - h_{box}^j}\right)^2$$  \hspace{1cm} (16)

$$\text{Class}_{\text{loss}} = \sum_{i=0}^{N} \sum_{j=0}^{B} \left[\|p_i(c) - \hat{p}_i(c)\|^2 + \sum_{i=0}^{N} \sum_{j=0}^{B} e_{ij}^c \right]$$  \hspace{1cm} (17)

$$\text{Confidence}_{\text{loss}} = \lambda_{\text{coord}} \sum_{i=0}^{N} \sum_{j=0}^{B} e_{ij}^c \left[(C_i - \hat{C}_j)^2\right]$$  \hspace{1cm} (18)

$$\text{Loss} = XY_{\text{loss}} + WH_{\text{loss}} + \text{Class}_{\text{loss}} + \text{Confidence}_{\text{loss}}$$  \hspace{1cm} (19)

where $N \times N$ represents the cell number of the feature map. $B$ represents the number of defects bounding boxes generated in each feature map. $e_{ij}^c$ represents whether there are defects in the cell $i$ and bounding box $j$, when there are defects in the bounding box $e_{ij}^c = 1$, otherwise $e_{ij}^c = 0$. $(x_i, y_i)$ represent the centre coordinates of the cell $i$, $(\hat{x}_i, \hat{y}_i)$ represent the actual centre coordinates of defects area. $\{w_{box}^i, h_{box}^i\}$ represent the defects bounding boxes size in the cell $i$, $\{w_{box}^j, h_{box}^j\}$ represent the actual sizes of the defects area. $p_i(c) - \hat{p}_i(c)$ is the difference between the probability of defects in the cell $i$ and the real situation. $C_i - \hat{C}_j$ is the confidence difference between the prediction bounding box and the truth bounding box. $\lambda$ is a variable constant.

### 3 Experimental results and discussion

#### 3.1 Datasets and evaluation criteria

In this section, typical fabric defects in production are selected as experimental samples, and automatic fabric defect detection equipment is built to collect fabric images to evaluate the fabric defect detection algorithm. The fabric defect detection experimental platform is shown in Fig. 6.

The components of fabric defect datasets are shown in Table 2, which contain two types of fabrics and five types of defects. The images of defects are shown in Fig. 3. From Figs. 3a-d, the linen fabric defects are classified as stain and broken course, stain, hole, linear scan CCD camera and machine vision light source.
The MAP is taken as the indicator to evaluate the accuracy of detection by constructing the multi-scale neural network. The learning rate is $10^{-4}$, the batch size is set to 16, and the bounding box threshold is set to 0.4. Also, the training set and testing set have been divided according to 8:2 before training; all algorithms have experimented with the same data. The fabric defects detection results with different methods are shown in Table 3.

According to these experimental results, it can be seen (i) compared with the SSD algorithm, the MSCNN has a higher accuracy of detection by constructing the multi-scale neural network and the accuracy of MSCNN is higher than fast R-CNN and faster R-CNN under the same network depth. (ii) The MSCNN we proposed has almost the same MAP as the Yolov3 algorithm, while the Yolov3 algorithm constructs a CNN with a depth of 53 layers DarkNet53. In terms of locating method, fast R-CNN and faster R-CNN uses a region proposal network (RPN) to locate defects. The SSD, Yolo and Yolov3 are using the bounding boxes generated randomly, which have no pre-known defects size to locate the defects.

To ensure the experiment fair, all the experiments are done carried out in the same environment: the program is compiled from Python3.6, the GTX1060 GPU is used to train and test detection models, 1000 iterations are carried out in the whole training process, the learning rate is $10^{-4}$, the batch size is set to 16, and the bounding box threshold is set to 0.4. Also, the training set and testing set have been divided according to 8:2 before training; all algorithms have experimented with the same data. The fabric defects detection results with different methods are shown in Table 3.

In this contrast experiment, we used randomly generated bounding boxes to locate the defects, to quantify the effect of defects bounding boxes with pre-known defects size on detection. The number of defects bounding boxes is set to 4, 8 and 12, which are named MSCNN-4, MSCNN-8 and MSCNN-12, respectively, and have the same network structure as MSCNN; the results are shown in Table 4. Similar conclusions can be drawn that the MSCNN tends to have better detection speed and accuracy by using defects bounding boxes with pre-known defects size to locate the defects in the process of detection.

Table 5 shows the detection accuracy of the proposed MSCNN algorithm of various defects on two types of fabrics. The results show that the accuracy of the proposed algorithm detection for defects are classified as hole, end out, stain, and double flat. All aspects: mean accuracy precision (MAP) and frames per second (FPS), the functions of precision and recall are shown as

$$\text{MAP} = \frac{1}{m} \sum_{i}^{m} \left( \int_{0}^{1} \text{Recall} \right) \text{precision} \quad (21)$$

where the ‘m’ means the types of defects, and the functions of precision and recall are shown as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (22)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (23)$$

where TP (true positives) represents the images correctly divided into positive examples, FPS (false positives) represents the images are negative but falsely divided into positive examples, FN (false negatives) represents the images are positive but falsely divide into negative examples. And the positive and negative represent whether there are defects in the final output defects bounding box.

### 3.2 Evaluation of defect detection performance

Fig. 7 MAP and FPS of experiment results with different detection method results on (a) Linen fabric, (b) Patterned fabric datasets

![Fig. 7](http://example.com/image.png)

| Methods       | MAP, % Linen fabric | Patterned fabric | FPS, s Linen fabric | Patterned fabric |
|---------------|---------------------|------------------|---------------------|------------------|
| fast R-CNN    | 60.4                | 65.7             | 0.9                 | 0.7              |
| faster R-CNN  | 82.1                | 85.6             | 13.7                | 13.6             |
| Yolo          | 70.4                | 71.2             | 20.1                | 20               |
| Yolov3        | 93.9                | 95.9             | 21.6                | 21.3             |
| SSD           | 89.3                | 91.2             | 23.1                | 22.7             |
| MSCNN         | 93.7                | 95.6             | 29.3                | 29.1             |

The evaluation criteria utilised in our experiments include two aspects: mean accuracy precision (MAP) and frames per second (FPS). The FPS means the number of images that can be detected within a second, which is taken as the indicator to evaluate the speed of detection, can be expanded as follows, and the frameCount means the number of detected images, the elapasedTime means the time of detecting these images

$$\text{FPS} = \frac{\text{frameCount}}{\text{elapasedTime}} \quad (20)$$

The MAP is taken as the indicator to evaluate the accuracy of detection, which represents the mean integral area of the precision–recall curve for each type of defects, the formula of MAP is shown as
Table 4  Comparison of detection results with different bounding boxes

| Methods    | Linen fabric | Patterned fabric |
|------------|--------------|------------------|
|            | MAP, %       |                  |
| MSCNN-4    | 86.3         | 89.2             |
| MSCNN-8    | 93.1         | 94.9             |
| MSCNN-12   | 93.4         | 95.1             |
| MSCNN      | 93.7         | 95.6             |
|            | Linen fabric | Patterned fabric |
|            | FPS, s       |                  |
| MSCNN-4    | 29.2         | 28.9             |
| MSCNN-8    | 23.7         | 23.4             |
| MSCNN-12   | 21.5         | 21.1             |
| MSCNN      | 29.3         | 29.1             |

Table 5  Defect detection accuracy of the proposed MSCNN algorithm for various fabrics

| Types          | Linen fabric | Patterned fabric |
|----------------|--------------|------------------|
|                | Stain, %     | End-out, %       |
| linen fabric   | 94.3         | 92.7             |
| patterned fabric | 97.4       | 94.6             |
|                | Broken curse, % | Double flat, % |
| linen fabric   | 93.1         | —                |
| patterned fabric | —           | 93.1             |
|                | Hole, %      |                  |
| linen fabric   | 94.9         |                  |
| patterned fabric | —           | 95.3             |

Table 6  Comparison of detection results of MSCNN with 26*26 scale feature neural network

| Types          | Linen fabric | Patterned fabric |
|----------------|--------------|------------------|
|                | Stain, %     | End-out, %       |
| linen fabric   | 93.9         | 86.4             |
| patterned fabric | 96.7       | 89.4             |
|                | Broken curse, % | Double flat, % |
| linen fabric   | 87.9         | —                |
| patterned fabric | —           | 81.3             |
|                | Hole, %      |                  |
| linen fabric   | 91.3         |                  |
| patterned fabric | —           | 93.9             |

Fig. 8  Detection results of linen fabric

Fig. 9  Detection results of patterned fabric

each kind of tiny defects is higher than 92%. This algorithm has good suppression ability for the wrong detection and missing detection.

To verify the influence of feature fusion on tiny scale fabric defects detection accuracy, we used MSCNN without a 13*13 feature map to do detection, which means the neural network only extracts 26*26 feature maps for detection. The results of detection accuracy are shown in Table 6. As we can see, for these tiny scale defects, such as end-out, broken curse and double-flat, there are obvious reduction of accuracy compared with Table 6, while the accuracy stain and hole defects are still higher than 90%. The conclusion can be drawn that the detection accuracy of tiny defects can be improved by using feature fusion to construct MSCNN.

3.3 Verification of fabric defect detection

The fabric defect image was input into the detection model to obtain detection results with classification and location, as shown in Figs. 8 and 9. The detection results include three kinds of information: defect types, defects bounding box and confidence. The defects bounding boxes of each colour correspond to different types of defects. The detection results show that even for the fabric with the complicated texture background and the overlapping region of defect features, it also can be detected precisely.

The FPS of the detection algorithm is 29.1 (29.1 images can be detected per second on average), the detection time required for each image is 30–40 ms. In the actual defect detection, the line-scan CCD camera's pixel is 4096*2, and capture a 4096*512 pixel fabric image, which is divided into eight 512*512 images input detection models for detection. The fabric width is 1.2 m, each detection of eight images is equivalent to the detection of fabric length of 0.15 m. The speed of detection can reach about 30 m/min, which is faster than manual detection.

4 Conclusion

This study proposed an MSCNN for fabric defect detection, which can improve the detection accuracy for tiny scale defects and the speed. Compared with traditional detection methods, MSCNN is able to extract different pixels of defects feature maps, which can enhance the representation of tiny scale fabric defects. Furthermore, this paper designed a faster location method that used the K-means clustering analysis obtaining defects bounding boxes with pre-known defects size information to replace the region generation method in location, realising the improvement of detection speed. In addition, the experiment results demonstrated that the proposed MSCNN realised the high detection accuracy for five common types of defects in linen fabric and patterned fabric, the detection accuracy for each type of tiny scale defects reached >92%, and the speed of detection reaches 30 m/min, which can satisfy the requirements of defect detection speed in production.

While there are still two limitations to the proposed methods. Firstly, this method is data-driven, which needs plenty of labelled fabric images to train the model. Secondly, due to the lack of fabric images, the fabrics with more complicated texture are not tested in this paper.

In the future, the few-shot learning method will be used in MSCNN to lower the demand for fabric images, which will use less data to train useful detection model. In addition, the more complicated fabrics and different structures of CNN (such as DenseNet and ResNet) will be investigated in further research to extend the types of fabrics we can detect.

5 References

[1] Kumar, A.: ‘Computer-vision-based fabric defect detection: a survey’, IEEE Trans. Ind. Electron., 2008, 55, (1), pp. 348–363
[2] Ngan, H.Y., Pang, G.K., Yung, N.H., et al.: ‘Automated fabric defect detection – a review’, Image Vis. Comput., 2011, 29, (7), pp. 442–458
[3] Mei, S., Wang, Y., Wen, G.: ‘Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model’, Sensors, 2018, 18, (4), p. 1064. Available at https://doi.org/10.3390/s18041064
[4] Tiloca, A., Borzone, P., Carosio, S., et al.: ‘Detecting fabric defects with a neural network using two kinds of optical patterns’, Tex. Res. J., 2002, 72, (6), pp. 545–550

[5] Zhou, J., Wang, J.: ‘Unsupervised fabric defect segmentation using local patch approximation’, J. Textile Inst., 2016, 107, (6), pp. 800–809

[6] Shi, M., Fu, R., Guo, Y., et al.: ‘Fabric defect detection using local contrast deviations’, Multimedia Tools Appl., 2011, 52, (1), pp. 147–157

[7] Zhu, D., Pan, R., Gao, W., et al.: ‘Yarn-dyed fabric defect detection based on autocorrelation function and GLCM’, Autex Res. J., 2015, 15, (5), pp. 226–232

[8] Harinath, D., Babu, K.R., Satyanarayana, P., et al.: ‘Defect detection in fabric using wavelet transform and genetic algorithm’, Trans. Mach. Learn. Artif. Intell., 2015, 3, (6), p. 10. Available at https://doi.org/10.4173/tmall.36.1551

[9] Li, P., Zhang, H., Jing, J., et al.: ‘Fabric defect detection based on multi-scale wavelet transform and Gaussian mixture model method’, J. Textile Inst., 2015, 106, (6), pp. 587–592

[10] Nguyen, H.Y., Pang, G.K., Yang, S.P., et al.: ‘Wavelet based methods on patterned fabric defect detection’, Pattern Recognit., 2005, 38, (4), pp. 559–576

[11] Escoset, J., Navarro, R.F., Millan Garcia-Verela, M.S., et al.: ‘Detection of local defects in textile webs using Gabor filters’, Proc. SPIE Int. Soc. Opt. Eng., 1998, 37, (8), pp. 163–170

[12] Li, M., Cui, S., Xie, Z.: ‘Application of Gaussian mixture model on defect detection of print fabric’, J. Text. Res., 2015, 36, (6), pp. 94–98

[13] Wei, B., Hao, K., Tang, X., et al.: ‘A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes’, Text. Res. J., 2019, 89, (17), pp. 3539–3555

[14] Li, Y., Zhang, D., Lee, D.J.: ‘Automatic fabric defect detection with a wide-and-compact network’, Neurocomputing, 2019, 329, pp. 329–338

[15] Yapi, D., Mejri, M., Allili, M.S., et al.: ‘A learning-based approach for automatic defect detection in textile images’, IFAC-PapersOnLine, 2015, 48, (3), pp. 2423–2428

[16] Ouyang, W., Xu, B., Hou, J., et al.: ‘Fabric defect detection using activation layer ensemble’, Multiconvolutional neural network’, IEEE Access, 2019, 7, pp. 70130–70140

[17] Yapi, D., Allili, M.S., Biazzni, N., et al.: ‘Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain’, IEEE Trans. Autom. Sci. Eng., 2018, 15, (3), pp. 1014–1026

[18] Jing, J.F., Ma, H., Zhang, H.H.: ‘Automatic fabric defect detection using a deep convolutional neural network’, Color. Technol., 2019, 135, (3), pp. 213–223

[19] Jing, J.F., Fan, X.T., Li, P.F., et al.: ‘Yarn-dyed fabric defect detection based on deep-convolutional neural network’, J. Text. Res., 2017, 38, pp. 68–74

[20] Wei, B., Hao, K., Tang, X., et al.: ‘Fabric defect detection based on faster RCNN’, Int. Conf. on Artificial Intelligence on Textile and Apparel, Hong Kong, 2018, pp. 45–51

[21] Liu, Z., Liu, S., Li, C., et al.: ‘Fabric defects detection based on SSD’. Proc. of the 2nd Int. Conf. on Graphics and Signal Processing, Sydney, Australia, 2018, pp. 74–78

[22] Ng, M.K., Ngan, H.Y., Yuan, X., et al.: ‘Patterned fabric inspection and visualization by the method of image decomposition’, IEEE Trans. Autom. Sci. Eng., 2014, 11, (3), pp. 943–947

[23] Li, X., Zhao, W., Pan, J., et al.: ‘Deformable patterned fabric defect detection with Fisher criterion-based deep learning’, IEEE Trans. Autom. Sci. Eng., 2017, 14, (2), pp. 1256–1264

[24] Jia, L., Chen, C., Liang, J., et al.: ‘Fabric defect inspection based on lattice segmentation and Gabor filtering’, Neurocomputing, 2017, 238, pp. 84–102

[25] Huanghuan, Z., Jinxiu, M.A., Junfeng, J., et al.: ‘Fabric defect detection method based on improved fast weighted median filtering and K-means’, J. Text. Res., 2019, 40, (12), pp. 50–56

[26] Chinei, L., Guangshuai, G., Zhourong, L., et al.: ‘Fabric defect detection based on biological vision modeling’, IEEE Access, 2018, 6, pp. 27659–27670

[27] Lin, T., Dollar, P., Girshick, R., et al.: ‘Feature pyramid networks for object detection’. Computer Vision and Pattern Recognition, Honolulu, USA, 2017, pp. 936–944

[28] Simonyan, K., Zisserman, A.: ‘Very deep convolutional networks for large-scale image recognition’. ICLR 2015: Int. Conf. on Learning Representations, San Diego, USA, 2015. Available at https://arxiv.org/abs/1409.1556

[29] Song, W., Wen, L., Guo, L., et al.: ‘Unsupervised fault diagnosis method based on iterative multi-manifold spectral clustering’, IET Collaborat. Intell. Manuf., 2019, 1, (2), pp. 48–55

[30] Zhang, Z., Chai, N., Liu I, Y., et al.: ‘Base types selection of PSS based on a priori algorithm and knowledge-based ANN’, IET Collaborat. Intell. Manuf., 2019, 1, (2), pp. 29–38

[31] Wuang, B., Li, W., Yang, W., et al.: ‘Illumination normalization based on Weber’s law with application to face recognition’, IEEE Signal Process. Lett., 2011, 18, (8), pp. 462–465

[32] Girshick, R.: ‘Fast R-CNN’. Int. Conf. on computer vision, Santiago, Chile, 2015, pp. 1440–1448

[33] Ren, S., He, K., Girshick, R., et al.: ‘Faster R-CNN: towards real-time object detection with region proposal networks’, Neural Inf. Process. Syst., Montreal, Canada, 2015, pp. 91–99

[34] Redmon, J., Divvala, S.K., Girshick, R., et al.: ‘You only look once: unified, real-time object detection’, Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 779–788

[35] Redmon, J., Farhadi, A.: ‘YOLO v3 An incremental improvement’, arxiv, 2018, ArXiv:1804.02767. Available at https://arxiv.org/abs/1804.02767

[36] Liu, W., Anguelov, D., Erhan, D., et al.: ‘SSD: single shot MultiBox detector’. European Conf. on Computer Vision, Amsterdam, The Netherlands, 2016, pp. 21–37