



Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization



Mengqi Chen^{a,b}, Lingjie Yu^{a,b,*}, Chao Zhi^{a,b,*}, Runjun Sun^{a,b}, Shuangwu Zhu^{a,b}, Zhongyuan Gao^{a,b}, Zhenxia Ke^{a,b}, Mengqiu Zhu^{a,b}, Yuming Zhang^{c,***}

^a School of Textile Science and Engineering, Xi'an Polytechnic University, Xi'an, Shaanxi 710048, China

^b State Key Laboratory of Intelligent Textile Material and Products, Xi'an Polytechnic University, Xi'an, Shaanxi 710048, China

^c Shaoxing University Yuanpei College, Shaoxing, Zhejiang 312000, China

ARTICLE INFO

Article history:

Received 9 June 2021

Received in revised form 9 September 2021

Accepted 22 September 2021

Available online 29 October 2021

Keywords:

Fabric defect detection

Faster R-CNN

Gabor filter

Genetic algorithm

ABSTRACT

Fabric defect detection plays a crucial role in fabric inspection and quality control. Convolutional neural networks (CNNs)-based model has been proved successful in various defect inspection applications. However, the sophisticated background texture is still a challenging task for fabric defect detection. To address the texture interference problem, taking advantage of Gabor filter in frequency analysis, we improved the Faster Region-based Convolutional Neural Network (Faster R-CNN) model by embedding Gabor kernels into Faster R-CNN, termed the Genetic Algorithm Gabor Faster R-CNN (Faster GG R-CNN); in addition, a two-stage training method based on Genetic Algorithm (GA) and back-propagation was designed to train the new Faster GG R-CNN model; finally, extensive experimental validations were conducted to evaluate the proposed model. The experimental results show that the proposed Faster GG R-CNN model outperforms the typical Faster R-CNN model in terms of accuracy. The proposed method' mean average precision (mAP) is 94.57%, compared to 78.98% with the Faster R-CNN.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Fabric defects occur during the textile production process and significantly affect the visual and mechanical properties of the final products; thus, the detection of fabric defects is a crucial part of quality control and inspection (Srinivasan et al., 1992). However, methods based on human vision and manual operation are unreliable due to visual fatigue, working state, and other subjective factors. Therefore, many fabric defect detection methods based on computer inspection have been developed in past decades (Ngan et al., 2011; Hanbay et al., 2016).

Fabric defect detection algorithms can be classified into two major groups: traditional image processing and Convolutional neural networks (CNNs)- based deep learning. The traditional image

processing approaches usually involve several stages, such as image transformation, feature extraction, and segmentation. Basically, the traditional image processing method can be sorted into the statistical, the structural, and spectral analysis (Song et al., 2015; Ye, 2009; Tsai and Huang, 2003). The statistical method involves such algorithms as co-occurrence matrix, histogram feature, autocorrelation function, and mathematical morphology (M. Li et al., 2019; Li et al., 2020). Raheja et al. (2013) extracted histogram features from the significant mapping of defective and defect-free fabrics to detect fabric defects. Zhang and Tang (2019) extracted texture features of fabric images based on gray level co-occurrence matrix (GLCM) by using sliding windows and then calculated texture energy, finally located defects according to energy deviation. The fabric structure is changeable, but certain rules can be obeyed (Zhang and Tang, 2019). Wang et al. (2020) segmented the image, extracted the texture feature of the defect image block, and identified the location of the defect through distance measurement and threshold segmentation. Jia et al. (2017) divided the fabric image into grids according to the change rules, compared the similarities between the lattices, then selected the flawless lattice to establish an average template as a unified reference for locating and correcting, and finally detected fabric defects. Spectral methods including wavelet transform, Fourier transform, and Gabor filtering are mainly used to remove

* Corresponding authors at: School of Textile Science and Engineering, Xi'an Polytechnic University, Xi'an, Shaanxi 710048, China.

** Corresponding author.

E-mail addresses: chenmengqi@stu.xpu.edu.cn (M. Chen), lingjie.yu@xpu.edu.cn (L. Yu), zhichao@xpu.edu.cn (C. Zhi), sunrunjun2002@163.com (R. Sun), zhushuangwu@263.net (S. Zhu), gaozhongyuan@stu.xpu.edu.cn (Z. Gao), kezhenxia@stu.xpu.edu.cn (Z. Ke), zhumengqiu@stu.xpu.edu.cn (M. Zhu), lunwenzhuanyou@126.com (Y. Zhang).

image texture. Vermaak et al. (2016) took the average energy of the real and imaginary parts of the complex wavelet coefficients as practical features respectively. They evaluated the performance of the Dual-Tree Complex Wavelet Transform (DTCWT) in fabric defect detection. The frequency and directional representation of Gabor filter are similar to that of the human visual system, thus often be used for texture filtering. As described in (Vermaak et al., 2016; Tong et al., 2016), both Gabor filtering and other methods were method used for defect detection. Through the experimental results of the above paper, it can be seen that the traditional image processing method is only effective for some defects with noticeable edges, single background, and relatively flat cloth.

The CNNs-based deep learning method has been adopted for automatic target detection in recent years (Zhang et al., 2020; Ouyang et al., 2019; Jing et al., 2017). Many detection models, including R-CNN (Girshick et al., 2014), SPPNet (He et al., 2016), Fast R-CNN (Girshick, 2015), Faster R-CNN (Xiang et al., 2020), all of which have a satisfactory performance in target detection application for many fields (Xiang et al., 2020; Y. Li et al., 2019).

Although neural networks have been proven to be effective in target detection in various areas, fabric creases, and sophisticated textures are inevitable in most fabric visual inspection occasions, causing massive disturbance to defect detection. Thus, in the task of detecting fabric defects, the interference of fabric background textures is a challenge, researches endeavor to improve the CNNs-based model (Li and Li, 2020; Jeyaraj and Samuel Nadar, 2019). Ouyang et al. (2019) introduced a novel pairwise-potential activation layer into CNNs to achieve high-precision defect segmentation on fabrics with complex features and unbalanced data sets. Li and Li (2020) improved Cascade R-CNN by adopting multi-scale training, prior frame clustering, and soft non-maximum inhibition, thus effectively raising the accuracy of fabric defect detection.

It has been proved that the texture pattern can be extracted by analyzing the frequency spectrum of fabric sample images (Zhang et al., 2020). In general, frequency characteristic analysis methods include Fourier transform, Wavelet transform, and Gabor transform. Among them, due to the merits of optimal joint localization in spatial and frequency domains, Gabor transform is regarded as a promising method for fabric defect detection (Tong et al., 2016; W. Li et al., 2012; Zhang et al., 2011). To filter the complicated surface texture, Hu (2015) proposed a new method based on the Genetic algorithm (GA) to optimize six Elliptic Gabor filter (EGF) parameters, including α , γ , u_0 , v_0 , F_0 , θ . W. Li et al. (2012) proposed the Log-Gabor filter with four parameters: σ_0 , θ_0 , σ_p , and ρ_0 to eliminate fabric textures. Based on the above researches, it can be concluded that the performance of Gabor filter heavily depends on the parameters setting of Gabor filter. Affected by the large variety of fabric textures (plain, twills with different angles and thicknesses, satin, et al.), the empirical parameters of Gabor filter cannot deal with different fabric defects situations without intelligent parameters optimization algorithms. Due to the excellent performance in parameter optimization, Genetic Algorithm (GA) attracts much attention. Tong et al. (2016) put forward the method of optimizing Gabor filter parameters by using composite differential evolution (CoDE) to achieve defects segmentation. D. Li et al. (2019) designed a new training method based on Multi-Population Genetic Algorithm (MPGA) and structure evaluation to optimize the Gabor kernel. In the process of GA, the optimized parameters can be achieved when the fitness function declares a maximum or minimum value (D. Li et al., 2019; Luo et al., 2020). The effectiveness of GA primarily relies heavily on a well-defined fitness function. In most researches, several empirical indicators such as the image entropy and gray-level differences between filtered image and defect-free image were adopted as the fitness function (Tong et al., 2016). However, in the fabric defect situation, too much filtering would lose the micro-defect information, whereas too light filtering would have little effect on reducing

texture interference. It is challenging for these empirical fitness functions to handle the delicate balance.

Based on the above discussions, a fabric defect detection model based on Gabor filter and neural networks was proposed in this paper. Due to the good performances in automatically target recognition for many public data sets, the Faster R-CNN was adopted as the neural networks' structures (Sun et al., 2019; Mudumbi et al., 2019; Xu et al., 2019). Furthermore, to eliminate the background texture, the Faster R-CNN was improved by embedding a Gabor filter as the first convolutional layer. We also adopt GA as Gabor parameters optimization algorithm. Moreover, the Average Precision (AP) value, which is the index of evaluating the accuracy of the detection model, was utilized to design the fitness function of GA. The main contributions of this paper are concluded as follows: (1) an optimized Gabor filter was introduced into the Faster R-CNN detection model, which helps to reduce the interference of fabric background; (2) GA was utilized to determine the optimal Gabor filter parameters; (3) different from empirical functions or middle indexes used in reported works, the final accuracy index of detection model was used to design the fitness function of GA (Roberge et al., 2018; Reddy et al., 2020; Sun et al., 2020), which helps to select the best Gabor filter parameters achieving the highest accuracy.

The rest of the paper is organized as follows. Section II introduces the theoretical knowledge of the Faster R-CNN model, Gabor filter, and GA, briefly; Section III explains the proposed network structures in detail; Section IV demonstrates the experimental results of the proposed model; Finally, Section V summarizes our work in this paper.

2. Theoretical background

2.1. Gabor filter

The frequency and direction representations of Gabor filters are close to those of the human visual system. Therefore, they are often used for texture representation and description. The texture analysis method based on Gabor filters can be divided into two sorts: a Gabor filter bank or an optimized Gabor filter. A multi-scale Gabor filter bank can improve detection accuracy whereas significantly increase computational cost. Considering the detection efficiency, we adopted one single Gabor filter with an optimized kernel in this paper. A 2-D Gabor filter is the product of a sinusoidal plane wave and a Gaussian kernel divided into the real part and the virtual part in the spatial domain. The 2-D Gabor filter definition can be expressed as follows:

$$G_{\lambda, \theta, \varphi, \sigma, \gamma}(x', y') = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \varphi\right)\right) \quad (1)$$

Filtering the real part $R_{\lambda, \theta, \varphi, \sigma, \gamma}(x', y')$ can make the image smoother whereas filtering virtual part $I_{\lambda, \theta, \varphi, \sigma, \gamma}(x', y')$ can help to detect the edge. The corresponding formula is showed as follows:

$$\begin{cases} I_{\lambda, \theta, \varphi, \sigma, \gamma}(x', y') = \exp[-(x'^2 + \gamma^2 y'^2)/2\sigma^2] \sin(2\pi x'/\lambda + \varphi) \\ x' = x \cos\theta + y \sin\theta \\ y' = -x \sin\theta + y \cos\theta \end{cases} \quad (2)$$

Where x and y represent the two directions (horizontal and vertical) of a pixel; λ and θ control the wavelength and orientation of the Gabor filters, respectively; σ is the Gaussian standard deviation of the filter; φ represents the phase; γ is the aspect ratio of space. The parameters of Gabor filters are relatively abstract, and the schematic diagram of filter kernel function with partial parameters is shown in Fig. 1.

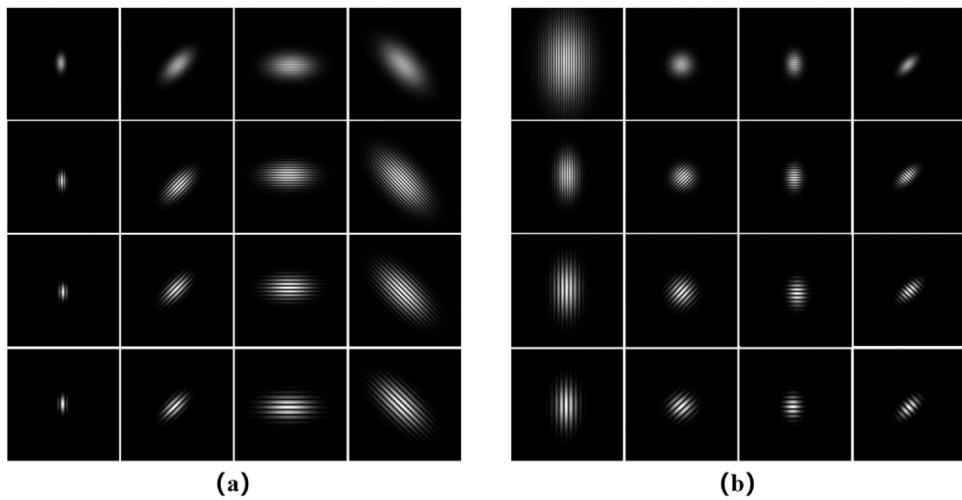


Fig. 1. The schematic diagram of Gabor filter with different parameters: (a) $\lambda = 5, 10, 15, 20$, $\theta = 0-180^\circ$, $\sigma = 5, 10, 15, 20$, $\gamma = 0.5$ (from left to right); (b) $\lambda = 5, 10, 15, 20$, $\gamma = 0.5, 1.0, 1.5, 2.0$, $\sigma = 15$, $\theta = 0-180^\circ$ (from left to right).

2.2. Genetic algorithm

GA was proposed inspired by natural biological selection. GA provides a general framework for searching global optimization parameters and ultimately produces optimal parameter combinations to solve problems, especially in the case of multiple parameters. Nature follows the principle of survival of the fittest, which is also the termination condition of the cross, selection, and mutation. Therefore, the definition of the fitness function is the key to GA. In this paper, the Gabor parameters were optimized through GA by continuous iteration with the evaluation index of Faster R-CNN be the fitness function.

The implement of GA includes the following:

2.2.1. Initial population

A GA starts with a population that represents a potential solution set to the problem. The population consists of a certain number of individuals that have been genetically coded. Each individual is an entity with the characteristics of chromosomes. In the process of parameter optimization, one chromosome represents one parameter. Therefore, we randomly generate three chromosomes as the initial population since three Gabor parameters need to be found.

2.2.2. Fitness function

The core of finding the optimal solution by GA is to design the fitness function. The fitness function will calculate the fitness of the chromosomes generated in this iteration, retain the chromosomes with high fitness, and remain the chromosome with the highest quality after multiple iterations.

2.2.3. Select operation

The parent and mother are selected for reproduction according to their fitness level. Thus, the principle of selection is that the fitter an individual is, the more likely it is to be chosen, thus continuously weeding out individuals with low fitness.

2.2.4. Cross and mutation

Genetic manipulation is performed on the selected parent body and mother body. Therefore, the offspring are produced by crossover, mutation, and other operators. On the basis of the copies of the parent body and mother body. Under the principle of retaining excellent genes largely, crossover increases the diversity of genes and improves the probability of finding the optimal solution.

2.3. Faster R-CNN model

CNNs exhibit good feature extraction and classification performance but low efficiency. Thus, Regions with CNNs features (R-CNN) was developed on the basic structure of CNNs with the improvement of transforming the to-be-detected target by Region Proposal method. Usually, the R-CNN series network recognizes the target by marking out the area of interest in the image. The process of a typical R-CNN model can be divided into three steps:

The first step is extracting regions from the feature map. Regional Proposal is an excellent region extraction method, that offers sliding windows with different widths and heights. Candidate target regions can be obtained through window sliding, then normalized to be the standard inputs of the next module;

The second step is extracting features from the input target regions, followed by convolution and pooling to obtain the outputs with fixed dimensions.

The last step can be divided into two stages, namely classification and boundary regression. The classification stage to classify the input vectors from the previous step (the classifier needs to be trained according to the characteristics). The boundary regression is aimed to obtain precise target areas through bounding box regression.

All the candidate target regions in R-CNN need to be extracted in advance, which is time-consuming and energy-costing. In addition, the input images of traditional CNNs should be normalized or cut to a fixed size, leading to target stretching or information losing. Aimed at the above problems in R-CNN, Faster R-CNN have been developed on the basic of R-CNN. Compared with R-CNN, Faster R-CNN introduces a Region of Interest (ROI) pooling layer to share the calculation work of features with original CNNs, and proposes the Region Proposal Network (RPN) to improve the efficiency of computation in region extraction.

3. Our method

3.1. Overview of our method

We develop a fabric defect detection model based on GA-optimized Gabor filter and Faster R-CNN networks to address the texture interference problem. In order to reduce edges from the fabric texture background and save computational cost, we introduced a Gabor filter to the Faster R-CNN as the first convolutional layer. For the proposed model, two groups of parameters need to be trained:

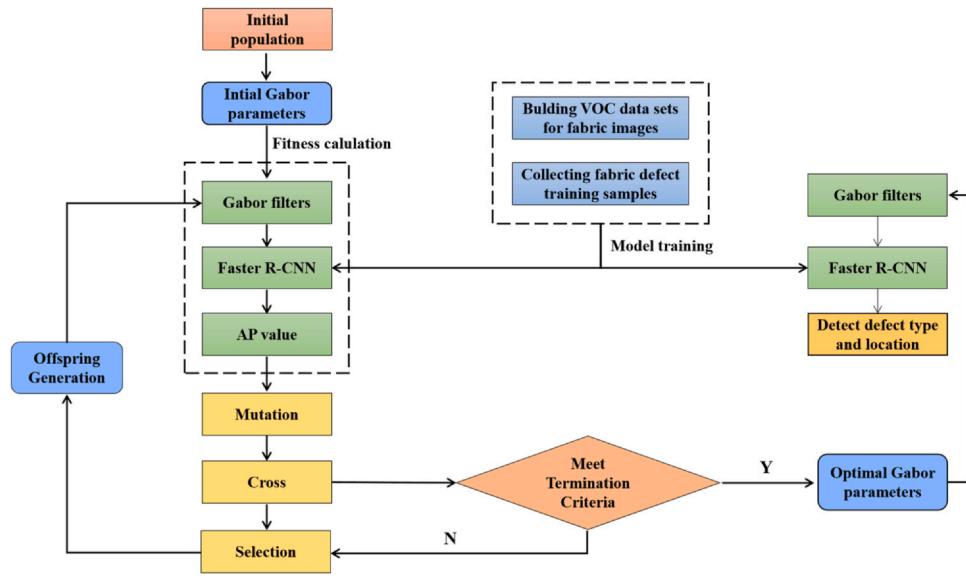


Fig. 2. The flowchart of the fabric defect detection process.

Gabor kernels and the weight matrix of the rest layers. Therefore, a two-stage training method was designed to train the whole model. In this method, the weight matrix of the convolutional and fully connected layers was trained by back-propagation. However, the Gabor kernels were trained through the optimization method based on GA instead of back-propagation. Since the Gabor layer has only three parameters and needs fewer iterations, the GA training would be much more energy-efficient. The flowchart of the proposed model was shown in Fig. 2.

3.2. Faster GG R-CNN structure

The structure of Faster GG R-CNN is displayed in Fig. 3. With a fabric defect image as the input, a Gabor filter layer was introduced as the first convolutional layer with parameters $(\theta, \lambda, \gamma)$ that need to be determined, followed by ResNet50 to produce the feature maps. The ResNet50 structure contains four different blocks. The first block

is illustrated by Conv_2 in Fig. 3, consisting of three convolutional layers with sizes of $1 \times 1 \times 64$, $3 \times 3 \times 64$, and $1 \times 1 \times 256$, respectively. Conv_3 is described as the second block, which contains 3 convolutional layers with sizes of $1 \times 1 \times 128$, $3 \times 3 \times 128$, and $1 \times 1 \times 512$, respectively. Finally, Conv_4 represents the third block, which includes 3 convolutional layers with sizes of $1 \times 1 \times 256$, $3 \times 3 \times 256$, and $1 \times 1 \times 1024$, respectively. Conv_5 is described as the fourth layer, which consists of 3 convolutional layers with sizes of $1 \times 1 \times 512$, $3 \times 3 \times 512$, and $1 \times 1 \times 2048$, respectively. In ResNet50, the four blocks were repeated three, four, six, and three times, respectively, as showed in Fig. 3. In addition, a single convolutional layer initiates the ResNet50, a fully connected layer ends the ResNet50; thus, a total number of 51 convolutional layers formed the backbone feature extraction networks. In this paper, a 2048-layer feature map was generated after the backbone feature extraction networks.

The Regional Proposal Network (RPN) is another essential module. The role of RPN is to generate proposals, including the

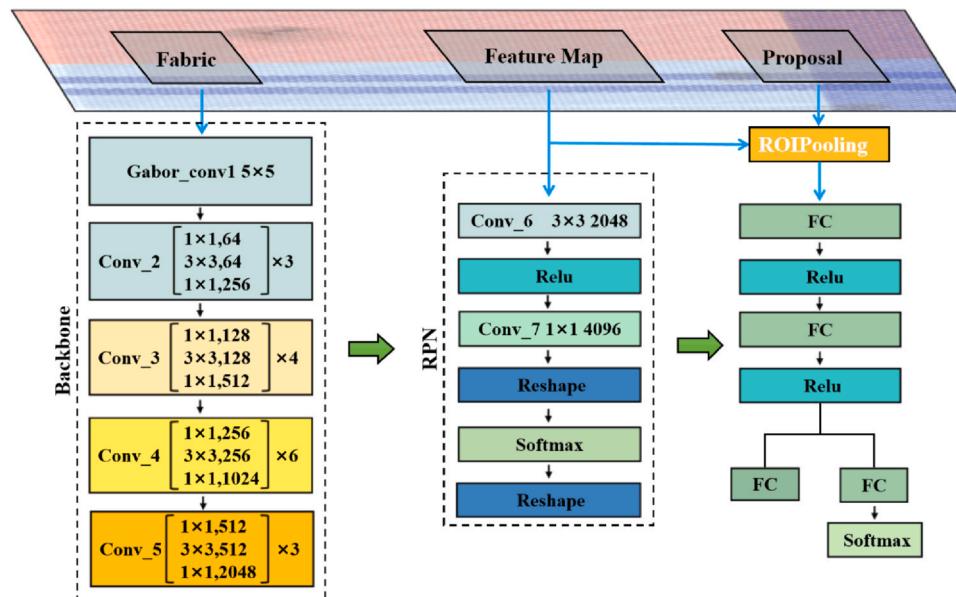


Fig. 3. The network structure of Faster GG R-CNN model.

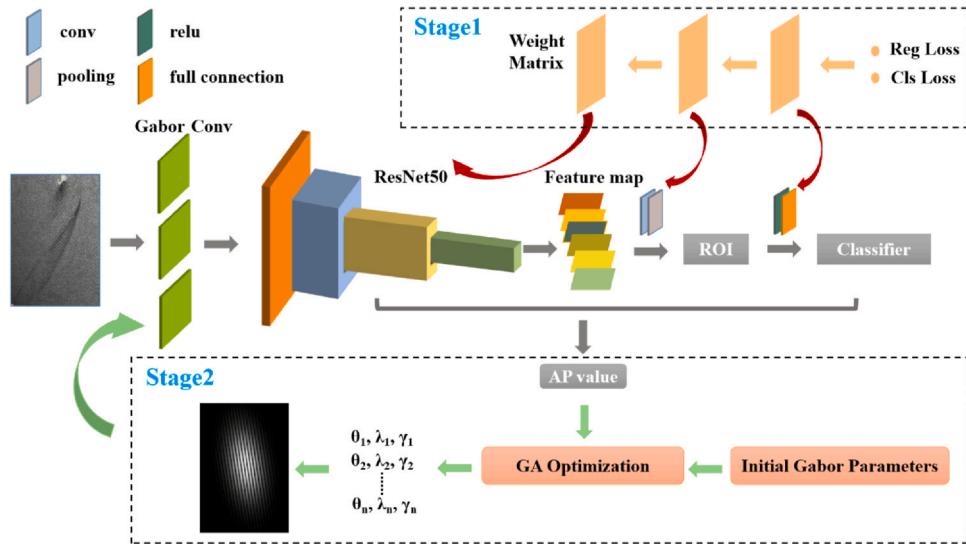


Fig. 4. Two-stage training method for Faster GG R-CNN model.

predicted boxes and possibilities for the proposals. With the generated feature map as the input, the RPN was implemented as a sliding window with $n \times n$ (in this paper, n denotes 3), scanning the whole feature map to generate multiple anchor boxes. RPN uses the Softmax layer to judge whether the anchor boxes positive or negative, namely the foreground or background. It is done by using 3×3 and 1×1 convolution layers to predict the scores belonging to the foreground or background, and the four regression coordinates, thus generating candidate regions, followed by Non-Maximum Suppression (NMS) operations to finetune and obtain the final region proposals. The ROI Pooling layer provided the decoding of the proposals. The Classification module, following the ROI Pooling, was composed of 3 full connected layers, 2 Relu layers, and a Softmax layer. Eventually, the final predicted results were marked by a rectangular box surrounding the defect targets. It is worth noting that the anchor boxes generated by RPN are different from the boxes locating the defect targets. The fabric image with a complicated background would increase the number of anchor boxes and proposals. Therefore, the existence of the Gabor layer, which could eliminate fabric texture, can help to narrow the number of anchor boxes and proposals and consequently reduce the calculation amount of the next Classification module.

3.3. Model training

The main energy-hungry steps of CNNs training (back-propagation) are gradient computation and the weight updates of the convolutional and fully connected layers (Meng et al., 2019). For the Gabor layer with only three uncertain parameters, it is unwise and inefficient to train Gabor kernels with back-propagation. To achieve energy efficiency, a simpler training method based on GA was designed to optimize the Gabor kernels of the proposed network.

The whole operation of model training was divided into two stages: (1) Training the network parameters except Gabor kernel based on back-propagation; (2) optimizing Gabor kernels based on GA. The illustration of the two-stage method is displayed in Fig. 4.

In the first stage, a pre-trained Gabor layer with parameters determined by an empirical fitness function was introduced as the first convolutional layer. Then, to train the network parameters except Gabor kernel, the gradient matrixes of whole full connection layers and the convolutional layers were built by the back-propagation method on the basis of the loss function. The parameters were then updated using the gradient matrixes. The loss function is

composed of the classification loss and the position regression loss, which can be described as follows:

$$L(\{p_i\}, \{t_i\}) = \sum_i L_{cls}(p_i, p_i^*)/N_{cls} + \lambda \sum_i p_i^* L_{reg}(t_i, t_i^*)/N_{reg} \quad (3)$$

Where i presents the subscript of each sample, p_i represents the probability of predicting the i -th anchor as a target, t_i is the 4D coordinate vector of the predicted position, t_i^* is the 4D position vector of ground truth, N_{cls} is the normalized size of cls item, λ/N_{reg} stands for the outside weights, N_{reg} expresses the number of anchor position normalized by reg item, and $L_{cls}(p_i, p_i^*)$ refers to classification loss. The classification loss is defined as $-\log [p_i^* p_i + (1-p_i^*) (1-p_i)]$, and p_i represents the probability of predicting the target as a specific category. p_i was set to be one if the current sample is positive and zero whereas the current sample is negative. p_i^* is the marked label of the real data. $L_{reg}(t_i, t_i^*)$ represents the border regression loss, which can be defined as:

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (4)$$

Where R refers to the $Smooth_{L1}$ function, which is calculated as:

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2, & |x| \leq 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (5)$$

The second stage is training the Gabor kernels (θ, λ, γ) out of the best combination. A fabric detect image was firstly input into a 5×5 Gabor layer and then sent to the rest pre-trained weight layers. Taking the network uniformity and final accuracy into consideration, we designed the fitness function of GA based on the evaluation index of the model - AP. AP refers to the area under the Precision-Recall Curve (PRC). Precision and Recall of object detection can be defined as:

$$\text{Precision} = TP/(TP + FP) \quad (6)$$

$$\text{Recall} = TP/(TP + FN) \quad (7)$$

Where TP (True Positive) is the number of samples that the detection model finds positive and indeed positive, FP (False Positive) refers to the number of samples that the detection model considers positive but actually negative, and FN (False Negative) indicates the number of samples that the detection model finds negative but actually positive.



Fig. 5. Defective fabric image samples.

4. Experiments

In this section, we introduce in detail the implementation of the proposed method on fabric defect samples. In addition, our tests were conducted on the Titan RTX GPU (24G) and TensorFlow (1.13.1) platform with Python 3.6.

4.1. Fabric defect image dataset

To help evaluate the performance of the proposed fabric defect detection method, a fabric defect dataset consisting of 6316 fabric image samples was collected and customized. The dataset was acquired from multiple resources, including apparel factories, literatures, and the “Xuelang Manufacturing AI Challenge” (Jing et al., 2017). The fabric defect dataset covered four types of defects: stains, holes, floats, and yarn defects, in which the yarn defects included broken weft, missing weft, double wefts, broken end, and missing end. All images were scaled to 500 pixel × 500 pixel. Color conversion and rotation were applied to the originally captured images to enlarge the dataset. The final images dataset size was enlarged to 6316, with 1341 hole defects, 1772 stain defects, 1599 float defects, and 1604 yarn defects. Among the whole dataset, 90% of samples were selected as training set and validation set, and the remaining 10% was the testing set. Parts of the samples are shown in Fig. 5.

4.2. Evaluation of fabric defect detection performance

The learning rate in the first stage of training was $1e^{-4}$ for the early 50 epochs and $1e^{-6}$ for the rest epochs. The epoch number was set to be 2000. The second training stage was mainly aimed to optimize the Gabor filters based on GA. The parameters contributing to the performance of GA are DNA size, POP size, crossover rate, mutation rate, and generations.

The value of DNA size is related to the range and precision of θ , λ , and γ in Gabor filters. The range of θ , λ , and γ is [0,180], [2,100] and [0,1], respectively, while the precision of the three kernels is 1, 0.03, and 0.001, respectively. Thus, 3266 values are needed to cover all

Table 1

The value range of GA parameters.

Parameter	Value range
POP size	[100,500]
Crossover rate	[0.4,0.99]
Mutation rate	[0.001,0.1]
Generations	[10,110]

possible selections of these three kernels, which is between 2^{11} and 2^{12} . Therefore, the value of DNA size was set to be 12. The normal value range of the other four GA parameters is shown in Table 1. To obtain and further justify the optimal parameters of GA, the uniform experimental design was adopted in this paper. Taking these four parameters (POP size, crossover rate, mutation rate, and generations) as the test factors, we designed the uniform design table with 11 levels and 4 factors. The uniform test design table of GA parameters $U_{11}(11^4)$ is shown in Table 2.

To what extend the GA can help to obtain the best defect recognition and the run time of GA are two main issues to be concerned about. Therefore, the mAP value of the detection model and GA's run time were both selected as the evaluation index to reflect the GA performance at different parameter combinations. The results of mAP and GA's run time obtained from the uniform experiments are also shown in Table 2. Furthermore, the quadratic polynomial stepwise regression model was adopted to analyze the uniform experimental results, which can be expressed as:

$$f(x) = \beta_0 + \sum_{i=1}^m \beta_i x_i + \sum_{i=1}^m \beta_{ii} x_i^2 + \sum_{i < j} \beta_{ij} x_i x_j \quad (8)$$

Where $f(x)$ is the function value of the index under different parameter combinations, x_i is the factor of the uniform design test, and i refers to 4 in this paper; x_1 , x_2 , x_3 , and x_4 represents POP size, means crossover, means mutation rate, and means generations, respectively; β_0 , β_i , β_{ii} , and β_{ij} are the regression coefficients; m is the number of independent variables which equals four in this paper; thus the regression model has $m(m + 3)/2$, namely 14 items in total.

Table 2The uniform test design table of GA parameters $U_{11}(11^4)$.

Level	POP size	Crossover rate	Mutation rate	Generations	Mean Average Precision (mAP)/%	GA's Run time/s
N1	380	0.90	0.025	60	92.79	16.19
N2	260	0.85	0.085	10	89.55	1.77
N3	500	0.75	0.055	40	92.75	13.62
N4	420	0.45	0.035	20	92.44	5.78
N5	100	0.55	0.095	50	93.31	3.40
N6	220	0.40	0.065	70	93.53	10.27
N7	340	0.70	0.105	80	93.40	18.18
N8	460	0.60	0.075	110	92.39	34.08
N9	300	0.50	0.005	90	92.47	18.04
N10	140	0.80	0.045	100	93.04	10.19
N11	180	0.65	0.015	30	90.93	3.74

Table 3

The significance test results of regression models.

Regression model	Test indices		
	R	R ²	F
Model based on mAP value	0.9998	0.9996	223.2953
Model based on GA's run time	1.0000	1.0000	55.555.4722

Table 4

Comparison of the two parameter combinations.

	Mean Average Precision (mAP)/%	GA's run time/s	Total runtime/s
X ₁ (x ₁ , x ₂ , x ₃ , x ₄)	94.57	10.79	78,790.79
X ₂ (x ₁ , x ₂ , x ₃ , x ₄)	91.49	1.74	78,781.74

Table 5

The value setting of parameters.

Parameter	Value or value range
DNA size	12
POP size	220
Crossover rate	0.9
Mutation rate	0.1050
Generations	70
θ	[0,180]
λ	[2,100]
γ	[0,1,1]

Table 6

Results of 5-fold cross-validation for Faster GG R-CNN model.

Fold	Average Precision (AP)/%			Mean Average Precision (mAP)/%	
	Hole	Float	Stain		
Fold 1	96.91	94.87	97.76	88.74	94.57
Fold 2	94.30	96.59	98.21	87.63	94.18
Fold 3	93.74	96.62	97.16	90.68	94.55
Fold 4	94.04	94.35	97.91	91.13	94.36
Fold 5	93.31	94.71	97.50	89.22	93.69

Table 7

The comparison of the detection model using back-propagation and GA.

Algorithm	Average Precision (AP)/%				mean Average Precision (mAP)/%	Training time/s
	Hole	Float	Stain	Yarn_defect		
Gabor improved Faster R-CNN using back-propagation	97.62	94.83	95.26	86.48	93.55	88,320.00
Gabor improved Faster R-CNN using GA (proposed)	96.91	94.87	97.76	88.74	94.57	78,790.79

The measured value of the regression coefficients was processed with DPS software. Using the mAP value and GA's run time as the evaluation index separately, two quadratic polynomial stepwise regression models of GA parameters were set up after the analysis of regression, which can be defined as:

$$f(x) = 94.831 - 17.357x_2 + 0.093x_4 + 2 \times 10^{-4}x_1^2 + 12.211x_2^2 - 9 \times 10^{-3}x_4^2 - 0.014x_1x_2 - 0.016x_1x_3 + 21.478x_2x_3 + 0.059x_2x_4 \quad (9)$$

$$f(x) = 3.902 - 6.478x_2 + 3 \times 10^{-6}x_1^2 + 6.705x_2^2 + 169.501x_3^2 + 5 \times 10^{-3}x_1x_2 - 0.007x_1x_3 + 10^{-1}x_1x_4 - 20.134x_2x_3 - 0.035x_3x_4 \quad (10)$$

Where (9) refers to the regression equation based on the mAP value, and (10) is the regression equation based on the run time of GA. To judge the significance of the two regression models, the significance tests were conducted, and Table 3 shows the significance test results of regression models. In Table 3, R, R^2 , and F represent the correlation coefficient, the coefficient of determination, and the significance level of the F test, respectively. Since the values of R and R^2 are both close to 1, and the F value is much greater than 0.05, the two regression models are both proven significant.

Based on the two regression models, two best parameter combinations of GA were obtained. The best parameter combination of GA taking mAP into consideration was expressed as $X_1(x_1, x_2, x_3, x_4)$ in which the POP size, crossover rate, mutation rate, and generations were set to 220, 0.9, 0.105, and 70, respectively, which could obtain the highest detection accuracy. The best parameter combination based on the GA run time was expressed as $X_2(x_1, x_2, x_3, x_4)$, in which the POP size, crossover rate, mutation rate, and generations were determined as 260, 0.5, 0.085, and 10, which could achieve the best detection efficiency. Table 4 exhibits a comprehensive comparison of the two parameter combinations. It can be seen from Table 4 that although the GA's run time of X_2 is shorter than that of X_1 , there is not much difference for the total model run time, since the GA's run time counts for a small share of the total run time of model. Furthermore, the detection accuracy is more worthy of concern,

making $X_1(x_1, x_2, x_3, x_4)$ the optimal parameter combination. The final value setting of parameters in the Gabor filter optimization is listed in Table 5.

The performance of the fabric defect detection algorithm was judged through four measurements, namely True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Precision and Recall can be calculated from the four measurements using (6) and (7). The indexes, Precision and Recall, are two restricted mutual indexes. Thus, it is prior to use one balanced index. On most occasions, the index AP is adopted to measure the comprehensive effectiveness of a detection model. By calculating the Precisions and Recalls of testing data, the Precision-Recall Curve can be created. AP value refers to the area under the Precision-Recall Curve. The mean Average Precision (mAP) reflects the average AP value, which is defined by the following.

$$mAP = \sum_{i=1}^N AP_i / N \quad (11)$$

Where AP_i indicates the AP value for each type, N refers to the number of types detected with the model.

To increase the accuracy of the results, the 5-fold cross-validation was performed. Table 6 indicates results of 5-fold cross-validation for Faster GG R-CNN model.

Since the main energy-consuming steps of CNNs training are gradient computation and the weight updates in the back-propagation process, GA was introduced to optimize the Gabor kernels of the proposed network instead of back-propagation. Therefore, a comparative experiment for the model using back-propagation and the proposed Genetic Algorithms was conducted to verify the effect of GA. Table 8 exhibits the comparison of the detection model using back-propagation and GA. It can be concluded from Table 7 that the mAP value of the proposed method is slightly higher than that of the model using back-propagation, while the proposed algorithm is more effective and time-saving.

Fig. 6 shows the examples of fabric defect detection results using the proposed Faster GG R-CNN network. The results indicate that the types and locations of different kinds of defects can be well detected. It can also be seen from this figure that the proposed method shows good performance in tiny defects and uneven fabrics with creases.

Table 8 exhibits the fabric defect detection results of four detection models. It can be seen from this table that for the mAP value, the proposed method is the best among the four models, indicating that the proposed method shows good comprehensive capability. However, it can also be found from this table that the proposed method performs the second-best concerning float defects. The possible reason is that the multi-scale features of M2Det are suitable for detecting feature pyramid structures, which makes M2Det more effective to detect objects with a large size, such as float defects.

Considering this method was proposed by improving the Faster R-CNN network structure, we also conducted the comparative experiments between the proposed method and the typical Faster R-DCNN model. Taking several fabric images with stain defects as examples, Fig. 7 illustrates the detection results from the Faster R-CNN network and the proposed method for the same pictures. It can be seen from Fig. 7 that the method based on the typical Faster R-CNN model tends to mistake fabric textures for stain defects, as shown in Fig. 7(a-2), Fig. 7(b-2), and Fig. 7(d-2). On the other hand, the method based on the Faster GG R-CNN network is more likely to detect the correct defect, as shown in Fig. 7(a-4) and Fig. 7(b-4), due to the advantage of Gabor Filter in effectively eliminating the fabric texture interference. Unfortunately, as shown in Fig. 7(d-4), our proposed method failed to recognize the color change region and mistake it for stain defect, as Faster R-CNN model did. Overall, the proposed method outperforms method based on Faster R-CNN for fabric defect detection.

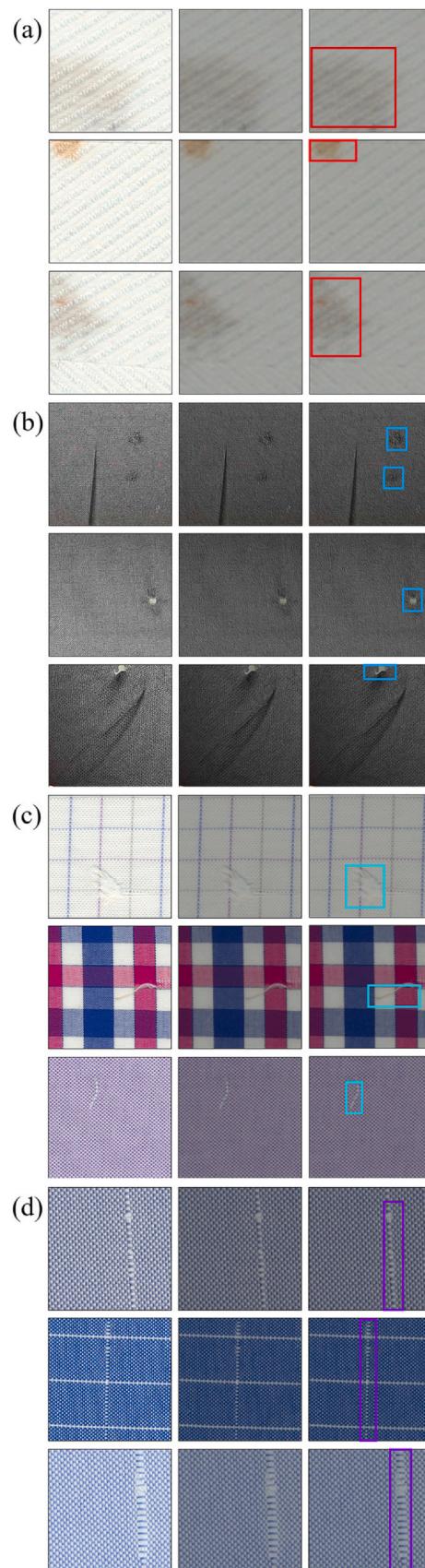


Fig. 6. Examples of fabric defect detection results using the proposed Faster GG R-CNN model: (a) detection results on the stain defect; (b) detection results on the hole defect; (c) detection results on the float defect; (d) detection results on the yarn defect. For each sub-image, the first column indicates the original fabric images, the second column illustrates the filtered images after Gabor conv, and the third column presents the detected defects.

Table 8

Fabric defect detection results of four detection models.

Algorithm	Average Precision (AP) / %				Mean Average Precision (mAP) / %
	Hole	Float	Stain	Yarn_defect	
Faster R-CNN	71.92	85.58	84.73	73.69	78.98
M2Det	91.06	96.29	94.56	80.79	90.67
Yolov3	82.80	84.49	76.93	75.48	79.92
The proposed Faster GG R-CNN	96.91	94.87	97.76	88.74	94.57

5. Conclusions

In this paper, an efficient fabric defect detection model was proposed based on Faster R-CNN. Firstly, by taking advantage of Gabor filter in texture eliminating, a Gabor kernel was introduced as the first layer into the Faster R-CNN networks with ResNet50 be the backbone. Secondly, the GA was adopted to select the best Gabor parameters. The fitness function was designed to achieve an optimal Gabor filter, which can help the whole model best recognize fabric defects. Finally, the verified experiments were carried out based on actual fabric samples. The detection results demonstrate that the proposed model can effectively recognize fabric defects of various backgrounds, locations, and sizes, even for tiny defects or uneven fabrics with creases. Comparative experiments were also conducted

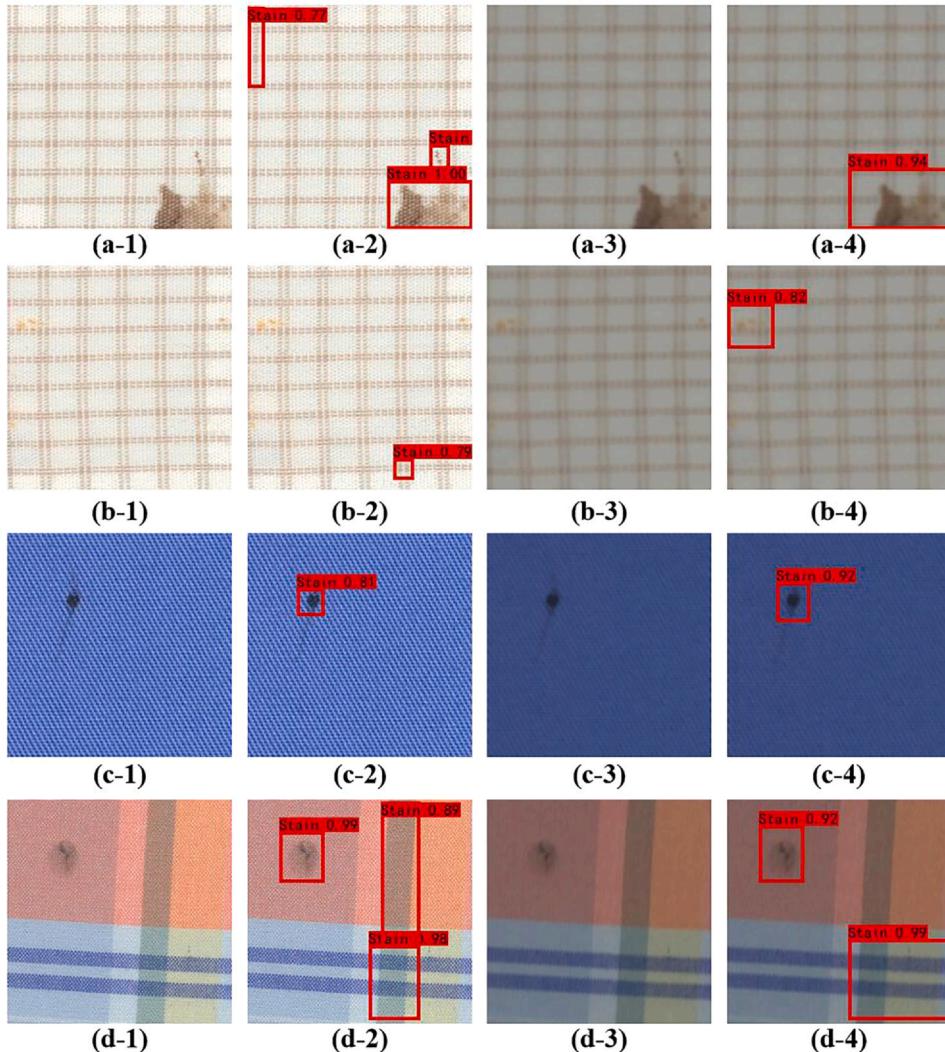


Fig. 7. The detection results from Faster R-CNN network and the proposed method for the same images. The first column indicates the original fabric images, the 2th column illustrates the detected defects by using the Faster R-CNN, the 3th column presents the filtered images after Gabor filter, and the 4th column displays the detected defects with the proposed method.

based on the proposed Faster GG R-CNN model and other three detection models. The performances of the methods were qualified with value AP. The results indicate that the proposed model has strong anti-interference capability and detection ability.

CRediT authorship contribution statement

Mengqi Chen: Methodology, Investigation, Software, Writing – original draft. **Lingjie Yu:** Conceptualization, Investigation, Writing – review & editing, Supervision, Funding acquisition. **Chao Zhi:** Methodology, Writing – review & editing, Funding acquisition. **Runjun Sun:** Resources, Validation. **Shuangwu Zhu:** Data curation. **Zhongyuan Gao:** Visualization. **Zhenxia Ke:** Validation. **Mengqiu Zhu:** Investigation. **Yuming Zhang:** Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the financial support from the National Natural Science Foundation of China (Grant No. 51903199), Outstanding Young Talents Support Plan of Shaanxi Universities (2020), Natural Science Basic Research Program of Shaanxi (No. 2019JQ-182 and 2018JQ5214), Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 18JS039), Science and Technology Guiding Project of China National Textile and Apparel Council (No. 2020044 and No. 2020046).

References

- Girshick, R., 2015. Fast R-CNN. In: 2015 IEEE Int. Conf. Comput. Vis., pp. 1440–1448.
- Girshick, R., Donahue, J., DarrellT., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, pp. 580–587. <https://doi.org/10.1109/CVPR.2014.81>.
- Hanbay, K., Talu, M.F., Özgürén, Ö.F., 2016. Fabric defect detection systems and methods—a systematic literature review. *Optik* 127, 11960–11973.
- He, K., ZhangX., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE, pp. 770–778.
- Hu, G.H., 2015. Automated defect detection in textured surfaces using optimal elliptical Gabor filters. *Optik* 126, 1331–1340.
- Jeyaraj, P.R., Samuel Nadar, E.R., 2019. Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm. *Int. J. Cloth. Sci. Technol.* 31, 510–521.
- Jia, L., Chen, C., Liang, J., Hou, Z., 2017. Fabric defect inspection based on lattice segmentation and Gabor filtering. *Neurocomputing* 238, 84–102.
- JingJ.F., DongA., LiP.F., 2017. Yarn-dyed fabric defect classification based on convolutional neural network. In: Ninth Int. Conf. Digit. Image Process. 56, 104202T.
- Li, D., Fang, H., Zheng, L., Ji, C., Yuan, H., 2019. Fabric defect detection algorithm for dense road and sparse road. IOP Conf. Ser.: Earth Environ. Sci. 252, 022077.
- Li, F., Li, F., 2020. Bag of tricks for fabric defect detection based on Cascade R-CNN. *Text. Res. J* 004051752095522.
- Li, F., Yuan, L., Zhang, K., Li, W., 2020. A defect detection method for unpatterned fabric based on multidirectional binary patterns and the gray-level co-occurrence matrix. *Text. Res. J*. 90, 776–796.
- Li, M., Wan, S., Deng, Z., Wang, Y., 2019. Fabric defect detection based on saliency histogram features. *Comput. Intell.* 35, 517–534.
- Li, W., Xue, W., Cheng, L., 2012. Intelligent detection of defects of yarn-dyed fabrics by energy-based local binary patterns. *Text. Res. J*. 82, 1960–1972.
- Li, Y., Zeng, J., Shan, S., Chen, X., 2019. Occlusion aware facial expression recognition using CNN with attention mechanism. *IEEE Trans. Image Process* 28, 2439–2450.
- Luo, X.L., Feng, J., Zhang, H.H., 2020. A genetic algorithm for astroparticle physics studies. *Comput. Phys. Commun.* 250, 106818.
- Meng, F., Wang, X., Shao, F., Wang, D., Hua, X., 2019. Energy-efficient Gabor kernels in neural networks with genetic algorithm training method. *Electronics* 8, 105.
- Mudumbi, T., Bian, N., Zhang, Y., Hazoume, F., 2019. An approach combined the Faster RCNN and mobilenet for logo detection. *J. Phys. Conf. Ser.* 1284, 0–8.
- Ngan, H.Y.T., Pang, G.K.H., Yung, N.H.C., 2011. Automated fabric defect detection – a review. *Image Vis. Comput.* 29, 442–458.
- Ouyang, W., Xu, B., Hou, J., Yuan, X., 2019. Fabric defect detection using activation layer embedded convolutional neural network. *IEEE Access* 7, 70130–70140.
- Raheja, J.L., Ajay, B., Chaudhary, A., 2013. Real time fabric defect detection system on an embedded DSP platform. *Optik* 124, 5280–5284.
- Reddy, G.T., Reddy, M.P.K., Lakshmanna, K., Rajput, D.S., Kaluri, R., Srivastava, G., 2020. Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. *Evol. Intell.* 13, 185–196.
- Roberge, V., Tarbouchi, M., Labonte, G., 2018. Fast genetic algorithm path planner for fixed-wing military UAV Using GPU. *IEEE Trans. Aerosp. Electron. Syst.* 54, 2105–2117.
- SongW., ChenF., GuZ., Gai, W., HuangW., WangB., 2015. Wood materials defects detection using image block percentile color histogram and eigenvector texture feature. In: Int. Conf. Inf. Sci. Mach. Mater. Energy, Chongqing, pp. 779–783.
- Srinivasan, K., Dastoor, P.H., Radhakrishnaiah, P., Jayaraman, S., 1992. FDAS: a knowledge-based framework for analysis of defects in woven textile structures. *J. Text. Inst.* 83, 431–448.
- Sun, X., Gu, J., Huang, R., Zou, R., Palomares, B.G., 2019. Surface defects recognition of wheel hub based on improved faster R-CNN. *Electron* 8, 1–16.
- Sun, Y., Xue, B., Zhang, M., Yen, G.G., Lv, J., 2020. Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Trans. Cybern.* 50, 3840–3854.
- Tong, L., Wong, W.K. 1, Kwong, C.K. 2, 2016. Differential evolution-based optimal Gabor filter model for fabric inspection. *Neurocomputing* 173, 1386–1401.
- Tsai, D.M., Huang, T.Y., 2003. Automated surface inspection for statistical textures. *Image Vis. Comput.* 21, 307–323.
- Vermaak, H., Nsengiyumva, P., Luwes, N., 2016. Using the Dual-Tree Complex Wavelet Transform for improved fabric defect detection. *J. Sens.* 2016, 1–8.
- Wang, W., Deng, N., Xin, B., 2020. Sequential detection of image defects for patterned fabrics. *IEEE Access* 8, 174751–174762.
- Xiang, J., Dong, T., Pan, R., Gao, W., 2020. Clothing attribute recognition based on RCNN framework using L-Softmax loss. *IEEE Access* 8, 48299–48313.
- Xu, Q., Zhang, X., Cheng, R., Song, Y., Wang, N., 2019. Occlusion problem-oriented adversarial Faster-RCNN scheme. *IEEE Access* 7, 170362–170373.
- YeY., 2009. Fabric defect detection using fuzzy inductive reasoning based on image histogram statistic variables. In: Int. Conf. Fuzzy Syst. Knowl. Discov., 6, pp. 191–194.
- Zhang, B., Tang, C., 2019. A method for defect detection of yarn-dyed fabric based on frequency domain filtering and similarity measurement. *Autex Res. J*. 19, 257–262.
- Zhang, K., Yan, Y., Li, P., Jing, J., Wang, Z., Xiong, Z., 2020. Fabric defect detection using saliency of multi-scale local steering kernel. *IET Image Process.* 14, 1265–1272.
- Zhang, Y., Yuen, C., Wong, W., Kan, C., 2011. An intelligent model for detecting and classifying color-textured fabric defects using genetic algorithms and the Elman neural network. *Tex. Res. J*. 81, 1772–1787.