Defect detection in textile fabrics with optimal Gabor filter and BRDPSO algorithm

Jiawei Zhang¹, Yueyang Li*¹ and Haichi Luo²

¹Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, 214122, China
²College of Internet of Things Engineering, Jiangnan University, Wuxi, 214122, China
*Corresponding author’s e-mail: lyueyang@jiangnan.edu.cn

Abstract. This paper presents an effective method that can detect fabric defects. The method utilizes the optimal Gabor filter and binary random drift particle swarm algorithm (BRDPSO) that can implement feature selection and parameter optimization synchronously. The parameters of 2D-Gabor filters are adjusted by quantum-behaved particle swarm optimization algorithm (QPSO) and the optimal Gabor filter is obtained. BRDPSO is used to select features on the original feature set and simultaneously optimize the parameters of the Isolation Forest (IF) classifier. Extensive experimental results indicate that the proposed method has effective detecting performance on the defect detection of textile fabric.

1. Introduction
Nowadays, the textile manufacturing industry employs the machines to handle the job for the large volume and rapidity, but the defection on the fabric could occur due to the error of the machine[1]. Traditionally, many textile enterprises use skilled inspectors to detect the defection. However, after investigation, the overall accuracy of visual inspection yielded no more than 80%[2]. Thus, various methods have constantly been developed to improve the efficiency and accuracy of textile fabric defect detection.

Adjustable Gabor filter can tune the parameters of a single filter through optimization method to obtain the filter parameters that are most suitable for the texture feature of flawless background image. This method reduces operation time as much as possible and has strong real-time performance, which is more suitable for industrial production[3]. In this paper, we use QPSO-based optimal Gabor filter to perform a convolution operation with the fabric image to smooth the background pattern and further highlight the defective area.

Feature extraction is the most important steps in the process of fabric defect detection. However, many literatures have shown that more features would not guarantee better performance. Redundant and irrelevant features will seriously affect the accuracy and efficiency of defect detection. Selecting effective features from the initial feature vector can reduce the computational complexity and improve the real-time detection of defects for subsequent identification work[4]. In addition, a large number of studies have proved that setting improper parameters will significantly reduce the performance of the classifier. Therefore, optimizing the parameters of the classifier is one of the effective ways to improve the accuracy of defect recognition[5].

Many scholars utilize two-step operation method, that is, firstly, reducing the dimension of feature vector, and then optimizing classifier model parameters[6]. However, this will increase the training
time. Meanwhile, feature selection and parameters optimization are not independent, but mutually influenced and related.

Based on the above problems, BRDPSO algorithm was creatively proposed in this paper with the aim of feature selection and parameters optimization synchronously, so as to ensure the real-time, accuracy and reliability of fabric defect detection.

2. Defect detection process and algorithms
The process of the proposed method is shown in Figure 1. Feature extraction is carried out on the filtered images convoluted with an optimal Gabor filter instead of the original fabric images. Then, BRDPSO algorithm, which can simultaneously select features and optimize classifier parameters, is proposed innovatively. Finally, the optimal feature subset and the optimal isolated forest classifier are used to detect the test images.

![Figure 1. Defect detection of textile fabrics process block diagram](image)

2.1. Optimal Gabor filter
A 2D-Gabor function is complex sinusoidal grating of given orientation modulated by a 2D-Gaussian function, which is given by equation (1):

\[
G(x, y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\}\exp\left\{j\left[2\pi Fx^* + \varphi\right]\right\}
\]

(1)

Where
\[
x^* = x \cos \theta - y \sin \theta
\]
\[
y^* = x \sin \theta + y \cos \theta
\]

\(\theta\) represents the radian direction of the Gabor function, \(F\) is the central frequency, \(\varphi\) expresses the phase shift, \(\sigma_x\), and \(\sigma_y\) is the standard deviation of Gabor function in x and y-axes. A 2D-Gabor filter is determined by a set of parameters \(\phi=(F, \theta, \sigma_x, \sigma_y, \varphi)\).

In general, the optimal Gabor filter parameters are selected so that the corresponding energy is a maximum for each specific texture. A defect-free texture image is selected as the template image to maximize its corresponding energy. For a given template image, the objective function is defined as in equation (2):

\[
fit = \frac{\mu(\phi)}{\sigma(\phi)}
\]

(2)

The mean \(\mu(\phi)\) and the standard deviation \(\sigma(\phi)\) are defined as in equation (3) and equation (4):

\[
\mu(\phi) = \frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} E_\phi(x, y)
\]

(3)
\[
\sigma(\phi) = \left( \frac{1}{N^2} \sum_{x=1}^{N} \sum_{y=1}^{N} \left[ E_r(x,y) - \mu(\phi) \right]^2 \right)^{1/2}
\]  

(4)

The energy response of the filter \( E_r(x,y) \) can be defined as in equation (5):

\[
E_r(x,y) = \left\{ \left[ T(x,y) \ast G_r(x,y) \right]^2 + \left[ T(x,y) \ast G_o(x,y) \right]^2 \right\}^{1/2}
\]

(5)

“\( \ast \)” is the mark of convolution operation. \( T(x,y) \) is the preprocessed flawless image. \( G_r(x,y) \) and \( G_o(x,y) \) are respectively the real part and the imaginary part of Gabor filter.

Within the specific solution space, to find the Gabor filter parameters that match the \( T(x,y) \) texture features, it is necessary to maximize the mean \( \mu(\phi) \) of the energy response value and minimize its standard deviation \( \sigma(\phi) \). That is to maximize the objective function \( f_{it} \).

2.1.1. QPSO algorithm. In this paper, the technique of QPSO algorithm is used to find the best parameters \( \phi \). QPSO algorithm is an improvement of PSO algorithm, which removes the motion direction attribute of particles, introduces the average optimal position of particles, and enhances the global convergence[7].

In the paper, a set of parameters of the Gabor filter \( \phi=(F, \theta, \sigma_r, \sigma_o, \phi) \) needs to be optimized, so the overall search space of the particle swarm (the total dimension of the particle swarm) can be set to \( D=5 \). The procedure of optimizing the parameters of the proposed method using QPSO is given by the following steps:

Step 1: Assume that the number of particles is \( M \) and the maximum number of iterations is \( Max\_iter \). At the initial moment, when \( t=0 \), the position of particle \( i \) is \( X_i(0)=\{X_{r,i}^{0}, X_{o,i}^{0}, \ldots, X_{o,i}^{0}\} \) and \( P_{best}(0)=X_i(0) \), where \( P_{best} \) is the best position of the particle;

Step 2: \( P_{best} \) is used to calculate \( M_{best} \), which is the average best position of the particle swarm.

\[ M_{best}(t+1) = \frac{1}{M} \sum_{i=1}^{M} P_{best,i}(t) \]  

(6)

Step 3: Evaluate the fitness value of each particle using equation (2);

Step 4: The way to update the optimal position of individual particles is shown in equation (7):

\[
P_{best}(t+1) = \begin{cases} 
X_i(t), & \text{if } f_{it}[X_i(t)] > f_{it}[P_{best}(t)] \\
P_{best}(t), & \text{if } f_{it}[X_i(t)] \leq f_{it}[P_{best}(t)]
\end{cases}
\]

(7)

The way to update the global optimal position of particles is shown in equation (8) and equation (9):

\[
g = \arg \max_{i=1}^{M} \left\{ f_{it}[P_{best,i}(t+1)] \right\}
\]

(8)

\[
G_{best}(t+1) = P_{best,g}(t+1)
\]

(9)

Step 5: Calculate the local attractor of each particle. In the iteration \( t \), the way to calculate the local attractor \( P_{a,i}(j=1,\ldots,5) \) of particle \( i \) is defined in equation (10):

\[
P_{a,i}(t+1) = r \times P_{best,i}(t+1) + (1-r) \times G_{best}(t+1)
\]

(10)

\( r \) is a random number that is uniformly distributed on the interval \( (0,1) \).

Step 6: Update the position of each particle. The position of particle \( X_{a,i}(j=1,\ldots,5) \) is updated by equation (11):

\[
X_{a,i}(t+1) = P_{a,i}(t) \pm \varepsilon \left[ M_{best}(t) - X_{a,i}(t) \right] \ln \left( \frac{1}{u_{a,i}(t)} \right)
\]

(11)
\( u_{ij}(t) \) is a random number that is uniformly distributed on the interval \((0,1)\). The probability of "+" or "-" is 0.5, \( \alpha \) can be controlled by linearly decreasing with the number of iterations.

Step 7: Repeat Step 2 to Step 6 until the maximum number of iterations is reached.

2.2. Select features and optimize parameters synchronously

In this paper, isolated forest (IF) algorithm is used as the classifier. BRDPSO algorithm is innovatively proposed to select effective features from the original feature set and synchronously optimize the classifier parameters.

2.2.1. Isolated forest classifier. The isolated forest (IF) algorithm is an efficient anomaly detection algorithm which detects outliers by isolating sample points. The isolated forest is composed of \( N \) trees, and the learning process of each tree is very random. It will randomly select samples and randomly select features to build an isolated tree, so that each sample is divided into an independent child node. Finally, we have \( N \) different isolated trees. From the perspective of hyperspace, sample points are continuously segmented with randomly selected hyperplanes until all sample points are "isolated" by these hyperplanes, that is, separated from other sample points. Points that use fewer hyperplanes can be isolated, that is, points that are particularly easy to be isolated will be judged as "outliers"[8].

2.2.2. BRDPSO algorithm. Random Drift Particle Swarm Optimization (RDPSO) algorithm is a PSO variant proposed by Jun Sun et al. It is inspired by the free electron model in a metal conductor placed in an external electric field[9].

In BRDPSO algorithm, each particle contains two parts, namely the isolated forest parameters part and the features part to be selected. Figure 2 shows the structure of each particle.

![Figure 2. The structure of each particle](image)

The first 4 decimal numbers of each particle represent the four parameters of IF algorithm. The last \( n \) bits binary string represents the characteristic mask. If \( K_i \) is "1", it means that the corresponding feature is selected, and vice versa, "0" means that the corresponding feature is not selected. The value \( F_i \) is determined by the dimension of the feature vector. The procedure of the proposed method using BRDPSO to select feature subset and synchronously optimize IF parameters is given by the following steps:

Step 1: Same as Step1 of QPSO optimization algorithm.

Step 2: The calculation method of \( M_{best}(j=1,\ldots,4) \) is the same as equation (6). Figure 3 shows the update method of \( M_{best}(j=5,\ldots,D) \), namely the binary string part of \( M_{best} \).

![Figure 3. The binary string part of \( M_{best} \) update method](image)
By counting the probability of occurrence of 0,1 for each bit of the binary code of the particle in the population, if there are many occurrences of 0, \( M_{\text{best}} \) corresponds to 0; If not, it is 1. If 0 and 1 appear the same times in the corresponding bit, then \( M_{\text{best}} \) randomly selects 0 or 1.

Step 3: Evaluate the fitness value of each particle. The fitness function \( f_{\text{it}} \) based on the specific feature selection method is shown by equation (12):

\[
    f_{\text{it}} = W_c \times AUC + W_f \times \left( \frac{1}{\text{ones}} \right)
\]

Where \( W_c + W_f = 1 \)

\( W_c \) is the classification precision weight and \( W_f \) is the feature subset weight. \( \text{ones} \) is the number of features selected in the feature subset. \( AUC \) score is an evaluation index to evaluate the classification model, which is defined as the area under ROC curve.

The fitness value of each particle is calculated according to the fitness function. According to the defect detection method, the maximal AUC score and the minimal number of the selected features lead to better performance. It means to maximize the objective function \( f_{\text{it}} \).

Step 4: Same as Step 4 of QPSO optimization algorithm.

Step 5: The calculation method of the local attractor, \( (1 \ldots, i \ldots, 4) \), the same as equation (10).

For the attractor \( P_{\text{best}}(j = 5 \ldots, D) \), it can be calculated by the cross operation of genetic algorithm. First, \( P_{\text{best}} \) and \( G_{\text{best}} \) are randomly divided into \( K \) parts, and then two new descendants are generated by crossover. Finally, one of the two descendants is selected randomly as the attractor \( P_{\text{best}} \). As shown in Figure 4, \( P_{\text{best}} \) and \( G_{\text{best}} \) are both composed of 8-bit binary strings.

\[
\begin{array}{cccccccc}
\text{Pbest} & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
\text{Gbest} & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
p^{1} & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
p^{2} & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
\end{array}
\]

Figure 4. The binary string part of \( P_{\text{best}} \) update method

Firstly, \( P_{\text{best}} \) and \( G_{\text{best}} \) were randomly divided into three parts. Each part is required to have at least 2 binary strings and at most 3 bits. The first three bits of \( P_{\alpha} \) are from \( P_{\text{best}} \), the third to the fifth bits are from \( G_{\text{best}} \), the last two bits are from \( P_{\text{best}} \). Using the same method, we can obtain \( P_{\beta} \). Finally, \( P_{\alpha} \) or \( P_{\beta} \) is randomly selected as the binary bit string part of the attractor \( P_{\text{best}} \) of the \( i \) particle.

Step 6: Update the position and velocity of each particle. The velocity of particle \( V_{\alpha} (j = 1 \ldots, D) \) is updated by equation (13):

\[
V_{\alpha}(t + 1) = \alpha \times |M_{\text{best}}(t + 1) - X_{\alpha}(t)| + \beta \times |p_{,\alpha}(t + 1) - X_{\alpha}(t)|
\]

where \( r \) is a random number that is uniformly distributed on the interval \((0,1)\). \( \alpha \) is drift coefficient and \( \beta \) is thermal coefficient. \( \alpha \) and \( \beta \) can be controlled by linearly decreasing with the number of iterations.

For the location of each particle, the first four and the last \( n \) dimensions are updated using different methods. The position of first four dimensions particle \( X_{\alpha}(j = 1 \ldots, 4) \) is updated by equation (14):

\[
X_{\alpha}(t + 1) = X_{\alpha}(t) + V_{\alpha}(t + 1)
\]

The position of the last \( n \) dimensions particle \( X_{\beta}(j = 5 \ldots, D) \) is updated by equation (15):
\[
X_{\alpha_j}(t+1) = \begin{cases} 
1, & r < \text{sigmoid}[V_{\alpha_j}(t+1)] \\
0, & \text{otherwise} 
\end{cases}
\] (15)

\(r\) is a random number that is uniformly distributed on the interval \((0,1)\).

Step 7: Repeat Step 6 until the maximum number of iterations is reached. At this time, the first four dimensions of \(G_{best}\) are the best IF model parameters, and the last \(n\) dimensions are the selected features, in which "1" means to select this feature, and "0" means not to select this feature.

3. Experiments and experimental results

In this experiment, fabric images with complex texture background are used. The performance of the algorithm was measured by area under the curve (AUC), detection time, and the number of selected features. AUC indicates the ability of the classifier to distinguish different categories. It is calculated by measuring the area under the receiver operating characteristics, the curve that shows a relationship between Sensitivity and Specificity.

3.1. Experiment process

The data in this work consisted of some fabric images with different complex texture background. The size of each image is 800 x 600 pixels. In the experiment, each image is segmented by overlapping, and the small images with the size of 160 x 120 pixels obtained after segmentation are taken as the sample data.

3.1.1. Processed images. Figure 5 shows four representative sample images and the images after processing.

![Processed images](image)

(a) representative original sample images

(b) preprocessed sample images

(c) Sample images after convolution with the optimal Gabor filter

Figure 5. Original images and processed images

The first row is four original representative sample images. For each type of fabric image having the same texture background, we train a BRDPSO-IF model discussed in previous sections. The training data and the testing data should have the same texture background but they could have different kinds of defects. The second row shows the corresponding images after pre-processing, such as grey scale processing, median filtering, and homomorphic filtering to enhance images and remove noise. The third row presents the images after convolution with the optimal Gabor filter. It is obvious that the background of fabric images becomes smoother and the defect area becomes more prominent.

3.1.2. Classification.

Table 1 shows the distribution of total sample images for training testing.
Table 1. The distribution of total sample images for training testing.

|         | Training set | Testing set |
|---------|--------------|-------------|
|         | Non-defect   | Defect      | Non-defect | Defect |
| Samples | 1039         | 84          | 754        | 108    |

Table 2 shows the comparison results between the proposed BRDPSO-IF model and (no-Gabor) BRDPSO-IF model without using optimal Gabor filter.

Table 2. Experimental comparison results with or without optimal Gabor filter.

|         | BRDPSO-IF | (no-Gabor) BRDPSO-IF |
|---------|-----------|----------------------|
| AUC (%) | 96.48     | 76.62                |
| Time(s) | 0.471     | 0.310                |
| Features(d) | 148 | 150                  |

It can be seen from table 2 that the method proposed in this paper can be effectively applied to the fabric images with complex texture background. After the convolution processing of the optimal Gabor filter, the features extracted from the convolution images can better represent the defect information of images. At the same time, experimental results confirm that, for fabric images with complex background, the defect detection performance is not good when features are extracted directly from the original image.

In the next experiment, the BRDPSO-IF model proposed in this paper was compared with IF model without feature selection and parameter optimization; BRDPSO-IF1 model with feature selection only; RDPSO-IF model with parameter optimization only; BRDPSO-IF2 model with feature selection and parameter optimization separately. Table 3 shows the comparison results of different models.

Table 3. Experimental results of different models.

|         | BRDPSO-IF | IF | BRDPSO-IF1 | RDPSO-IF | BRDPSO-IF2 |
|---------|-----------|----|------------|----------|------------|
| AUC (%) | 96.48     | 89.64 | 93.11 | 92.81    | 93.76      |
| Time(s) | 0.471     | 0.546 | 0.862  | 0.541    | 0.494      |
| Features(d) | 148 | 309 | 154  | 309      | 154        |

It can be seen from table 3 that only about half of the features are retained, the average AUC score value is increased by 2.72%~6.84%. The detection time is also decreased by 0.023s~0.391s, which indicates that there are redundant features in the original feature set. Comparison of experimental results shows that BRDPSO algorithm can not only select useful features and shorten detection time, but also optimize classifier parameters and improve the accuracy of defect detection. In general, the BRDPSO-IF model proposed in this paper has the better performance than the models mentioned above.

4. Conclusion

In the experiment, the convolution operation of the optimal Gabor filter is innovatively added in the image pre-processing step, which makes it easier to distinguish the defect area from the background area. Meanwhile, in order to synchronize feature selection and parameter optimization, binary random drift particle swarm (BRDPSO) algorithm is innovatively proposed and it is combined with isolated forest (IF) algorithm to construct BRDPSO-IF model to detect defects of textile fabric images. The comparison experiment for the fabric images with complex texture background demonstrates that the
method is effective and robust, and the detection process is fast enough to be applied to industrial production.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (51405198). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal used for this research.

References
[1] Bumrungkun P. Defect detection in textile fabrics with snake active contour and support vector machines[J]. Journal of Physics: Conference Series, 2019, 1195: 012006.
[2] T. C R. Automated visual inspection techniques and applications: A bibliography[J]. Pergamon, 1982, 15(4).
[3] Li Y, Luo H, Yu M, et al. Fabric defect detection algorithm using RDPSO-based optimal Gabor filter[J]. The Journal of The Textile Institute, 2018, 110(4): 487-495.
[4] Yi-Quan W, Hong W, Zhi-Long Y. Feature Extraction of Fabric Defects Based on Complex Contourlet Transform and Principal Component Analysis[J]. Journal of Donghua University(English Edition), 2013, 30(04): 282-286.
[5] Liu S Y, Zhang L D, Wang Q, et al. BP neural network in classification of fabric defect based on particle swarm optimization[C]. International Conference on Wavelet Analysis & Pattern Recognition, 2008.
[6] Liu J, Hui J. Fuzzy Inspection of Fabric Defects Based on Particle Swarm Optimization (PSO)[M]. 2008.
[7] Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior[C]. Congress on Evolutionary Computation, 2004.
[8] Liu F T, Ting K M, Zhou Z. Isolation Forest[C]. 2008 Eighth IEEE International Conference on Data Mining, 2008: 413-422.
[9] Sun J, Wu X, Palade V, et al. Random drift particle swarm optimization algorithm: convergence analysis and parameter selection[J]. Machine Learning, 101(1-3): 345-376.