

RESEARCH

Open Access



# Fabric4show: real-time vision system for fabric defect detection and post-processing

Huaizhou Lin<sup>1</sup> , Dan Cai<sup>1\*</sup> , Zengmin Xu<sup>2,3\*</sup> , Jinsong Wu<sup>4,5</sup>, Lixian Sun<sup>1</sup> and Haibin Jia<sup>1</sup>

## Abstract

The exploration of computer vision applications for fabric defect detection has immense potential value. However, current relevant research in this area has primarily focused on detection models that aim for high detection accuracy and algorithmic efficiency, while neglecting the practical industrial production requirements. Therefore, we propose a fabric defect detection and post-processing system that integrates an optimized region with convolutional neural network (CNN) features (i.e., Faster R-CNN) for defect detection, defect localization and detection model evaluation. In addition, the proposed intelligent system incorporates novel approaches, such as a rearranged fabric dataset, anomaly detection, recommended clipping region division, and a replenishment device. This study illustrates an example of artificial intelligence (AI)-driven automated technology in fabric manufacturing. The accuracy and detection speed of different detection models under identical hardware conditions are evaluated and compared with related work. Experimental results demonstrate that the proposed approach achieves comparable performance to other models, while significantly reducing computational resource requirements. The potential efficiency of using two-stage networks on hardware systems for fabric defect detection tasks is highlighted, which is likely to have relevant implications for the textile industry.

**Keywords:** Computer vision, Fabric manufacturing, Automation, Defect detection, Faster R-CNN

## 1 Introduction

With the rapid development of society and economy, the demand for high-quality garments is increasing. As a result, fabric defect detection has become a critical step in production. However, in industrial manufacturing, the detection of many different small and weak target defects remains a challenge.

Moreover, fabrics with too many defects can only be sold at 45% ~ 65% of their normal price [1, 2]. If serious problems are not detected at the factory stage, the entire batch of products will be rendered unusable, which is detrimental

to textile enterprises. Therefore, the implementation of an efficient, reliable, and integrated intelligent vision system for the textile industry can result in significant economic benefits. Today, the textile industry requires a more specific set of criteria for fabric defect detection, such as a more comprehensive scoring system beyond counting the number of defects or recognizing their type, and a more reasonable post-processing strategy based on different defect distribution patterns. Many researchers have made consistent efforts to address these issues and have proposed excellent detection approaches [3–9].

However, the implementation of automatic intelligent control in textile enterprises still faces several problems. First, training a detection model is difficult because there are many different fabric defects but few available samples, which can ultimately lead to higher detection errors. Second, the post-processing strategies for defect detection need to be improved, as enterprises are concerned with how to effectively handle and categorize the detected de-

\* Correspondence: [dancai1985@guet.edu.cn](mailto:dancai1985@guet.edu.cn); [xzm@guet.edu.cn](mailto:xzm@guet.edu.cn)

<sup>1</sup>School of Materials Science and Engineering, Guangxi Key Laboratory of Information, Guilin University of Electronic Technology, Guilin, China

<sup>2</sup>School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Center for Applied Mathematics of Guangxi (GUET), Guilin, China

Full list of author information is available at the end of the article

fective fabrics to maximize their residual value. Finally, the lack of mature and comprehensive examples of real-world deployments makes it difficult for enterprises to determine the feasibility and cost of intelligently transforming their production lines.

Our contributions are summarized as follows.

1) We propose a fabric defect detection system that integrates an optimized region with convolutional neural network (CNN) features (i.e., Faster R-CNN) to improve the detection accuracy.

2) We design a comprehensive scoring mechanism based on the type, area, and direction of fabric defects, and present a defective fabric post-processing scheme based on orbital conveyor transfer and stencil replenishment.

3) We develop an anomaly detection mechanism based on defect continuity evaluation. If the detection system detects a large area of continuous defects, or if the detection value exceeds the alarm threshold, a stop command is sent to the production line control system, and an audible visual alarm is triggered.

4) We propose an algorithm for automatically recommending a clipping region division scheme, which effectively addresses the current reliance on manual segmentation for the reuse of defective fabric.

5) We have built a low-cost, real-time vision system for fabric defect detection and post-processing, and developed the supporting industrial software.

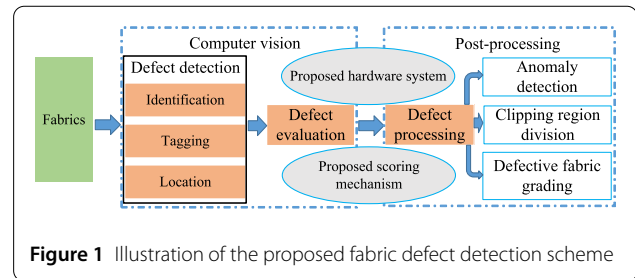
## 2 Related work

### 2.1 Research progress in fabric defect detection

In 2016, Hanbay et al. [10] provided a detailed overview of fabric defects and proposed seven methods to detect them. Traditional machine vision methods [11–13] and the emission spectra and statistical methods have been used to identify defective areas and count defects prior to cutting the fabric. However, these methods have difficulty detecting random changes in fabric surface texture and are easily affected by lighting conditions, resulting in low accuracy in industrial environments.

In recent years, the development of graphics processing units (GPUs) has led to significant advances in fabric defect detection based on deep learning techniques. For instance, YOLO [14], Faster R-CNN [15], and other models and algorithms have not only achieved better performance but also have the ability to identify defect types and their spatial locations compared to traditional machine vision methods. However, deep learning algorithms require powerful GPUs, which increases the difficulty and cost of their deployment in practical industrial applications.

To balance costs and benefits, researchers have to reconsider the use of low-cost and rapid computer vision technology in practical industrial applications. Khodier et al. [16] used a combination of CNNs to detect defects in jacquard fabrics with complex patterns and achieved an



**Figure 1** Illustration of the proposed fabric defect detection scheme

accuracy of approximately 99%. Wang et al. [17] proposed a double-branch parallel Faster R-CNN model for multi-category defect detection in practical applications, which could handle a variety of fabric defects and achieved a mean of average precision (mAP) of 0.574 (mAP@50).

### 2.2 Challenges and opportunities

Although there have been significant advances in fabric defect detection, traditional machine vision methods are still widely used in the textile industry [18–20]. Manufacturers are increasingly demanding fabric defect detection systems that not only count the number of defects but also evaluate and process defective fabrics, which can be used to guide improvements in the manufacturing process. These requirements pose new challenges for vision technology, while offering new opportunities for the development of intelligent manufacturing.

To address these challenges, we design a comprehensive evaluation standard based on defect type, area, and direction, as shown in Fig. 1, and propose an algorithm that can automatically recommend region divisions for clipping and grading the defective fabric. This standard is also used for anomaly detection. All these functions are adjusted to meet the real needs of enterprises and have been successfully deployed in hardware modules.

## 3 Detection model

We report three design improvements based on the classic Faster R-CNN [21].

1) Multi-source datasets. We primarily use open-source datasets, including the smart diagnosis of cloth flaw dataset from Tianchi [22, 23] (main dataset), the fabric defect dataset from Kaggle [24, 25] (fine-tuned dataset A) and the additional images of fabric with defects acquired in this work (fine-tuned dataset B). Based on the cluster analysis results of defect size and cause, the original 30 categories are merged into 14 categories to simplify the detection process, which is also a common practice in related work [15, 17].

2) Double-stream training (DST). To detect small targets more efficiently, ResNet50 [26] is used in the backbone neural network instead of VGG-16. Unlike the scheme proposed by Wang et al. [17], both the mainstream and substream network of the Faster R-CNN are trained on the

main and fine-tuned datasets, respectively, to compute a mixed score to effectively balance the detection effects of large and small defects. CioULoss [27] is used to replace L1Loss to improve the detection efficiency of defects with different aspect ratios. In addition, NMS [28] is employed to efficiently remove highly overlapping predicted boxes with relatively inaccurate calibrations, 16-bit floating point (FP16) [29] is utilized to reduce GPU memory consumption, and online hard example mining (OHEM) [30] is applied to improve the model's ability to learn from difficult cases. Various loss functions are attempted to enhance the performance of the detection model.

3) Deep learning application of model output. The outputs of the detection model, such as the type, bounding box (bbox) position, and the confidence level, are utilized for further processing. Using the bbox position, the area and direction of each defect are first calculated and incorporated into the proposed scoring mechanism. Then, based on this scoring mechanism, anomaly detection and clipping region division are proposed and implemented. These algorithms and the hardware components are integrated into the proposed system.

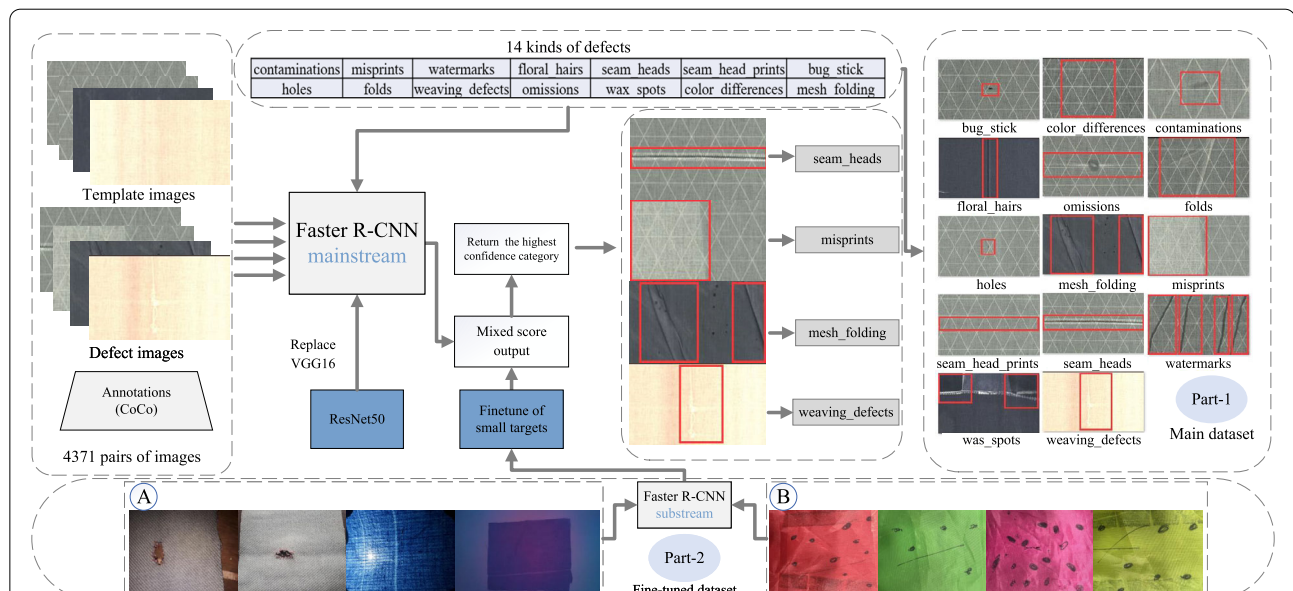
The workflow of the proposed fabric defect detection is shown in Fig. 2. The main dataset contains many pairs of images. In each pair, there are fabrics with the same pattern design. In one image, the fabric has defects, while in the other image, the fabric has no defects. We first use

the mainstream network of the Faster R-CNN to train our model, and then use the substream network of the Faster R-CNN to train the model with small defects in the fine-tuned dataset. Finally, the rating results of the two models are mixed, and the defect type with the highest confidence value is returned. In addition, Fig. 3(a) displays the number of defects contained in the dataset and Fig. 3(b) shows the comparison between the two datasets, the main dataset and the fine-tuned dataset.

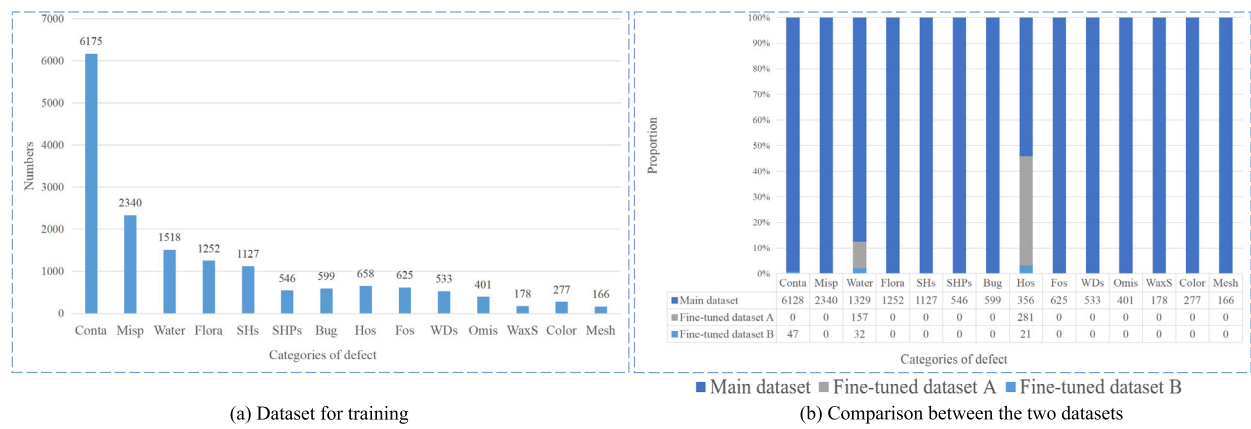
#### 4 Fabric defect scoring mechanism

From a technical perspective, training an artificial intelligent (AI) application model is only one step in the engineering process. Enterprises are not usually concerned with the type of detection models (Faster R-CNN, Cascade R-CNN or YOLO) used. In contrast, they are more concerned with having a feasible and cost-effective solution that includes hardware, software, and detection methods.

The training of a fabric defect detection model is considered a pre-task for system deployment. Applying the model with the industrial camera and the optimal use of the information returned by the detection models are areas that need to be explored. After all, the lack of accuracy of single shot detection can be compensated for by multiple shots. This shows that efficient and reasonable processing schemes are more important than high single shot detection accuracy in real industrial production. Therefore, this



**Figure 2** The workflow of the proposed fabric defect detection method and the rearranged fabric dataset in this work. For the main dataset, we integrate multiple fabric defect datasets [22, 23] into a new dataset. The fine-tuned dataset A consists of short lines and holes from the Kaggle's fabric dataset [24, 25]. The fine-tuned dataset B consists of contaminations and short watermarks collected in this work. The CoCo stands for common objects in context. It is a dataset format that includes annotated images with object segmentation masks, bounding boxes, and labels. The VGG16 stands for visual geometry group 16-layer, the Faster R-CNN is the faster region-based convolutional neural network, and the ResNet is the residual network



**Figure 3** Details of our dataset. Conta: contaminations; Misp: misprints; Water: watermarks; Flora: floral\_hairs; SHs: seam\_heads; SHPs: seam\_head\_prints; Bug: bug\_stick; Hos: holes; Fos: folds; WDs: weaving\_defects; Omis: omissions; WaxS: wax\_spots; Color: color\_differences; Mesh: mesh\_folding

work proposes a scoring mechanism that provides criteria for post-processing, which is introduced in subsequent sections.

In actual production, enterprises do not always discard or recycle a defective piece of fabric. It is more economical to adopt different strategies depending on the specific situation of the defect distribution. Therefore, we have designed a comprehensive scoring criterion to evaluate the type, area, and direction (transverse or longitudinal) of fabric defects. This criterion is used as the basis for further processing of the defective fabric.

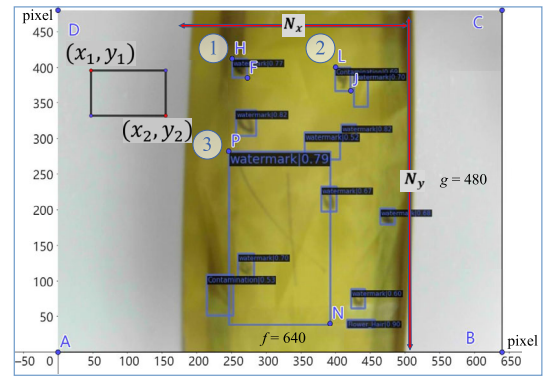
The total score of the fabric defects is denoted as  $P^{\text{Tot}}$  and calculated as follows:

$$\begin{aligned}
 P^{\text{Tot}} &= \sum_{i=1}^n P_i \\
 &= \sum_{i=1}^n (\alpha P_{i,\text{typ}} + \beta P_{i,\text{are}} + \gamma P_{i,\text{tra}} \\
 &\quad + \delta P_{i,\text{lon}}),
 \end{aligned} \quad (1)$$

where  $n$  represents the total number of detected defects,  $P_i$  denotes the score of the  $i$ -th detected defect, while  $P_{i,\text{typ}}$ ,  $P_{i,\text{are}}$ ,  $P_{i,\text{tra}}$ , and  $P_{i,\text{lon}}$  are the scores in terms of type, area, transverse and longitudinal trend, respectively.  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\delta$  are the weight coefficients, whose values are negatively related to the acceptance of the corresponding defect characteristics and can be adjusted by the analytical hierarchy process or neural network according to different customer needs.

#### 4.1 Single defect scoring

The coordinates returned by the detection model for positioning the bbox are  $(x_1, y_1)$ ,  $(x_2, y_2)$ , as shown in Fig. 4.



**Figure 4** An example of scoring fabric defects based on bounding box (bbox) location. The coordinates returned by the detection model for positioning the bbox are  $(x_1, y_1)$ ,  $(x_2, y_2)$ .  $f$  and  $g$  indicate the resolution ratio of current image.  $N_x$  and  $N_y$  represent the transverse and longitudinal lengths, respectively. We use the bbox position to calculate the area and direction of each defect, and use the data to help us evaluate the defect. Three bounding boxes in the figure represent three defects in the fabric. For the first bounding box's position,  $H = (251, 412)$ ,  $F = (273, 385)$ ; for the second bounding box's position,  $L = (400, 400)$ ,  $J = (422, 367)$ ; for the third bounding box's position,  $P = (246, 282)$ ,  $N = (392, 40)$ . In addition,  $A = (0, 0)$ ,  $B = (640, 0)$ ,  $C = (640, 480)$ ,  $D = (0, 480)$

The output of the  $i$ -th detected defect is recorded as a vector ( $D_i$ ):

$$D_i = [\text{type}_i, x_{1,i}, y_{1,i}, x_{2,i}, y_{2,i}], \quad (2)$$



which is used to calculate the width  $\Delta x_i$  and length  $\Delta y_i$  of the defect as follows:

$$\begin{cases} \Delta x_i = |x_{2,i} - x_{1,i}|, \\ \Delta y_i = |y_{2,i} - y_{1,i}|. \end{cases} \quad (3)$$

The transverse trend score  $P_{i,tra}$  and longitudinal trend score  $P_{i,lon}$  of the defect can be obtained by normalizing its width and length. The industrial camera used in the proposed system has a direct output resolution ratio of  $640 \times 480$ , but the fabric does not occupy the entire industrial camera's field of view. Thus, the direct resolution ratio is calculated using Eq. (4):

$$\begin{cases} P_{i,tra} = \Delta x_i / N_x, \\ P_{i,lon} = \Delta y_i / N_y, \end{cases} \quad (4)$$

where  $N_x$  and  $N_y$  represent the transverse and longitudinal lengths, respectively. In addition, the score  $P_{i,are}$  of the defective area can be calculated using Eq. (5):

$$P_{i,are} = \frac{\Delta x_i \cdot \Delta y_i}{N_x \cdot N_y} = P_{i,tra} \cdot P_{i,lon}. \quad (5)$$

#### 4.2 Video frame scoring

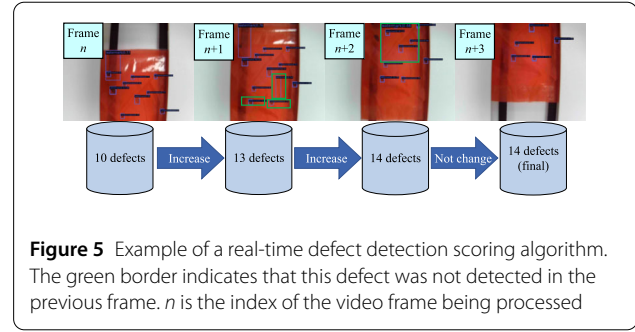
As shown in Fig. 4, the scores of all defects, including the three marked defects (1, 2, 3) in the current frame, are calculated by Eq. (1).

However, this calculation is limited to individual frames of the video stream. In actual production, the area of the fabric exceeds the field of view of the industrial camera, requiring adaptation algorithms. Therefore, we propose a real-time scoring algorithm to calculate the total score  $P^{Tot}$  for the entire fabric.

#### 4.3 Real-time fabric scoring mechanism

Our algorithm is based on frame-by-frame detection and scoring thresholds in ideal cases. However, practical industrial applications require a real-time detection system that can process successive frames of camera video, as the fabric on the conveyor belt moves in a direction relative to the industrial camera. In addition, the consistency of defect detection is limited to the detection model.

To solve these problems, we develop a mechanism using defect numbering and dictionary traversal. Defects are individually numbered and combined with other information to create a dictionary. Using this algorithm, the dictionary of defects between two adjacent frames is compared in real time to determine the duplication and loss of detected defects. Furthermore, the real-time scoring algorithm is combined with defect scoring and start/stop judgments to form a scoring mechanism. An example of this real-time scoring algorithm is shown in Fig. 5. By coupling this scoring mechanism with the hardware system



described in the following sections, defective fabrics are evaluated more accurately.

#### 4.4 Anomaly detection

On the production line, an abnormal condition of a knitting machine can cause major defects in the fabric. If left undetected, it can cause significant losses to the enterprise. To address this issue, we develop a more efficient scoring algorithm and an anomaly detection method based on the defect scoring density and the mechanisms introduced in the previous section. First, after defect scoring, the defective area is evaluated; in particular, the total score  $P_k^{Tot}$  ( $k$  represents the  $k$ -th fabric) and the corresponding defect score density  $\rho_k$  (also shown in Fig. 6) are calculated for each frame. When the following conditions are satisfied, a shutdown command is sent to the control system.

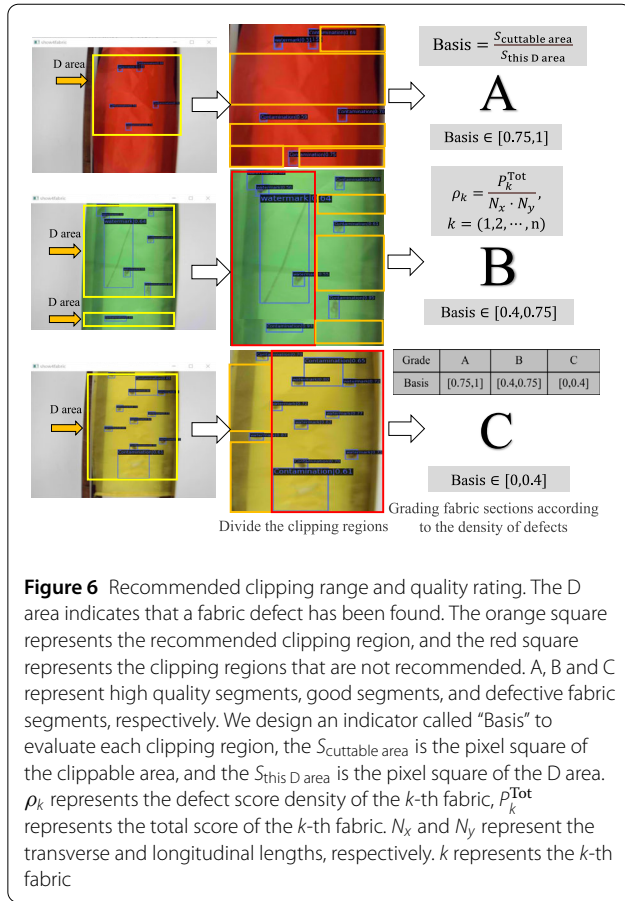
- 1) The defect score density exceeds the preset threshold
- 2) Any length, width, or area of a single defect exceeds its preset thresholds.

If the shutdown condition is not triggered after the timeout, the loop waits for the next detection area.

#### 4.5 Clipping region division

The distribution of defects in different parts of a long piece of fabric on the production line can vary. To address this issue, inspired by previous work [31–33], we design algorithms that recommend the division of clipping regions and evaluate the quality of each part, which can help enterprises maximize the economic value of the fabric. The experiment demonstrated the algorithms using only full-color red, yellow, and green fabrics, considering both point-like defects, such as contaminations, and line-like defects, such as watermarks, which are simpler and easier to detect with greater accuracy (Fig. 6).

When the industrial personal computer (IPC) provides the source instruction for defect processing, the system divides the fabric based on whether defects are present. Regions with defects are marked with yellow borders, while regions without defects are marked with blue borders. The previously mentioned scoring mechanism is used to provide ratings for the areas with defects. The irregular cutting of fabric is a non-deterministic polynomial (NP)-hard

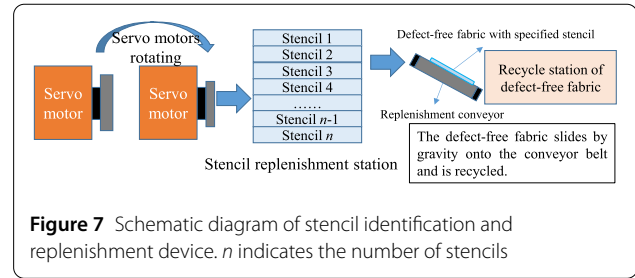


problem [31]; it is more practical and faster to cut and divide the fabric into rectangles in actual production. Therefore, the division of regions is simplified by solving the maximum rectangular area problem. We use orange borders to indicate recommended clipping regions and red borders to indicate non-recommended clipping regions.

Next, the system evaluates each clipping region of the fabric using different thresholds, where A, B, and C represent high quality segments, good segments, and defective fabric segments, respectively. Then, the partition information of the current fabric and the partition screenshot are logged for further classification processing.

#### 4.6 Stencil identification and replenishment

To ensure efficient and continuous fabric production, defective fabrics must be quickly replaced with the defect-free fabrics in the same stencil to avoid disrupting the stacking sequence [33]. This section describes the stencil identification and replenishment device, which includes a stencil replenishment station, a steering device, deflectors of the stencil replenishment station, a replenishment conveyor, and controllers capable of identifying stencil of fabrics as they move. For fast fabric pattern recognition, current methods use simple classification tasks



through neural network training [34], or pattern recognition through RGB histogram-based approaches [35]. The latter approach is used in this work, because providing additional models after fabric detection models will hinder the detection efficiency and speed of the system. An operation diagram of the stencil identification and replenishment device is shown in Fig. 7. Based on this approach, the corresponding execution logic in this paper is designed as follows.

1) When the defect score exceeds the set threshold, the IPC takes a screenshot of the camera video stream, compares its RGB channel histogram with the pre-stored pattern, calculates, and returns the ID of the pattern with the highest similarity.

2) The hardware system then releases the defect-free fabrics with the specified stencil in order as compensation, according to the identified pattern of the detected fabric.

#### 5 Structure and hardware

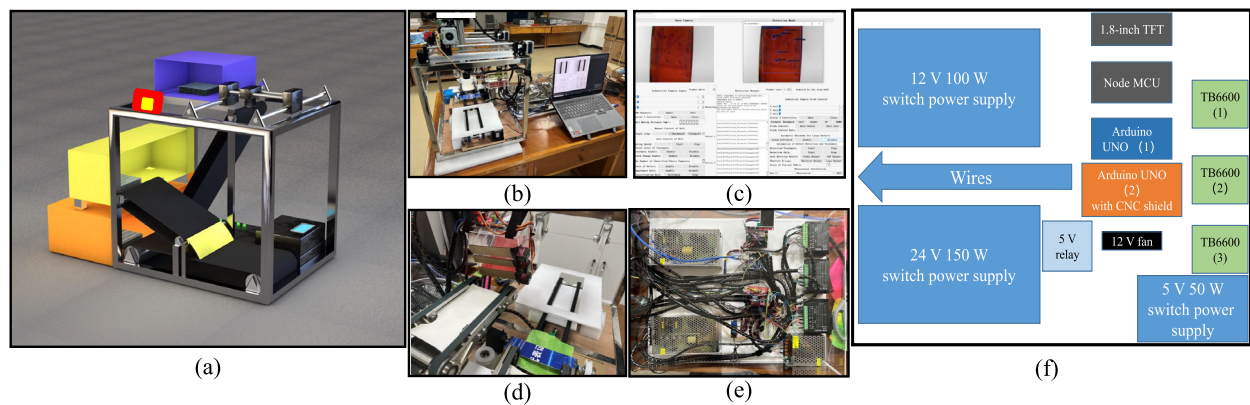
Existing research on fabric defect detection has focused mainly on software implementation. Although some studies have proposed a semi-manual system [18], there has been limited exploration of hardware-based fabric defect detection and post-processing systems [36]. In industrial applications, models, software and hardware are equally important. Therefore, a fabric defect detection and post-processing system will be designed and applied in practice.

Fig. 8 shows the overall rendering effect of the system, a physical image of the system in the main view, a schematic diagram, and close-up photographs of the hardware module, and a close-up photograph of the fabric replenishment module and the user interface of the IPC program.

The electrical components and modules of the proposed system are shown in Fig. 9 and Table 1. It consists of eight major components: IPC, main processing, expansion and display, drive execution, power, alarm, fabric replenishment, and vision modules.

1) The IPC module, which is used to run the fabric defect detection model, the IPC program, and to communicate with the hardware system, consists of a Lenovo Legion R9000P personal laptop with an R7-5800H CPU and an NVIDIA GeForce RTX 3060 Laptop (6 GB) GPU.

2) The main processing module, which is responsible for organizing all the components of the hardware system,



**Figure 8** Overall structure of our fabric defect detection and post-processing system. (a) Structural schematic diagram; (b) Real-time image of the system in the main view; (c) User interface of the industrial personal computer program; (d) Close-up photo of the fabric replenishment module; (e) Close-up photo of the hardware module; (f) Schematic diagram of the hardware module. The TFT is the thin-film transistor, the MCU is the micro-controller unit (the Node MCU is an ESP8266 based development board), the Arduino UNO is a product of Arduino. TB6600 is a commonly used stepper motor driver, which is used to control the movement of stepper motors



**Figure 9** Electrical devices and modules of the fabric defect detection and post-processing system. The IPC is the industrial personal computer

**Table 1** Electrical devices and modules in the proposed system. “DIY” means that the device is self-developed

Module	Device	Model	Amount
IPC module	Lenovo Legion R9000P	R7-5800H, RTX 3060 6 GB laptop	1
Main processing module	Development board	Arduino UNO R3 (ATmega328P)	2
	Extended board	CNC Shield V3	1
Extension and display module	Development board	Node MCU V3 WIFI (ESP8266)	1
	1.8-inch TFT screen	ST7735	1
Drive execution module	Stepper motor	42BYGH47	2
	Small conveyor belt	DIY	2
	Motor driver for stepper motor	TB6600	3
	Motor driver for sliding platform	A4988	3
	Light-sensitive sensor	LM393	1
Power module	Switching power supply	MW S-150-24 (150 W 24 V)	1
		MW S-100-12 (100 W 12 V)	1
		MW S-50-5 (50 W 5 V)	1
Alarm module	Alarm light	LTE-5061	1
	Relay	FL-3FF-S-Z 5 V DC	1
Fabric replenishment module	Servo control board	PCA9685	1
	Servo motor	MG90	3
Vision module	Industrial camera	HW500 (640×480)	1
	3D sliding platform	DIY	1

consists of two Arduino UNOs and a CNC Shield extended board.

3) The extension and display module, consisting of a Node MCU V3 WIFI board with ESP8266 and a 1.8-inch TFT screen of model ST7735, is used for human-machine interaction and communication extension.

4) The drive execution module contains two stepper motors of model 42BYGH47, two small conveyor belts we built ourselves, three motor drivers for the stepper motor of model TB6600, three motor drivers for the sliding platform of model A4988 and a light-sensitive sensor of model LM393. This module is used to receive control commands from the main processing module and drive the motors.

5) The power module provides different voltage requirements, using three power supply units: MW S-150-24 (150 W 24 V), MW S-100-12 (100 W 12 V), and MW S-50-5 (50 W 5 V).

6) The alarm module consists of an LTE-5061 alarm light and an FL-3FF-S-Z 5 V DC relay, providing hardware support for anomaly detection algorithms.

7) The fabric replenishment module consists of a PCA9685 servo controller board and three MG90 servo motors (which can be rotated from 90 degrees to 180 degrees). This module provides hardware support for stencil identification and replenishment for the post-processing session.

8) The vision module includes a HW500 industrial camera (5 million pixels, 30 frames per second, 640 × 480 direct output resolution ratio) and a self-developed 3D sliding platform. The industrial camera can capture moving images and centralize them via the controller and IPC.

## 6 Experiments and discussion

### 6.1 Experimental configuration

The rearranged fabric dataset used in this study is divided into main and fine-tuned datasets. Each part is further divided into training, verification, and test sets in the ratio of 6:2:2. The fine-tuned Faster R-CNN is used for training. The details of the training methods can be found in the introduction for detection models, and the training environment is as follows: Python 3.7.6, PyTorch 1.11.0+cu115, CuDNN 8.3.2, TorchVision 0.12.0+cu115, OpenCV 4.5.2, MNCV 1.4.8, and MMDetection 2.23.0.

The detection and post-processing system is deployed on a personal laptop with an R7-5800H CPU and an NVIDIA GeForce RTX 3060 Laptop (6 GB) GPU. With the CPU-assisted computation enabled, the system processes five frames per second of real-time detection video. Moreover, a monitoring software is developed using QT5 and connected to the hardware system via a serial port. The user interface of this software includes various functions, such as industrial camera video stream output, defect detection video stream output and industrial camera RGB adjustment, as shown in Fig. 8(c).

### 6.2 Model evaluation

#### 6.2.1 Performance comparison

The improved model is validated on a rearranged dataset and evaluated using the mean average precision (mAP). The proposed model achieves promising performance, especially in reducing the GPU memory cost (during model training) and improving the small target detection accuracy, as shown in Table 2. Compared to those of the baseline, the values of mAP@0.5:0.95, mAP@0.5, and



**Table 2** Ablation experiment of the proposed method. mAP@0.5:0.95, mAP@0.5, mAP@0.75, mAP\_s@0.5:0.95, mAP\_m@0.5:0.95, and mAP\_l@0.5:0.95 represent the mAP values at different thresholds. For example, mAP@0.5:0.95 represents the average mAP value calculated at intersection over union (IoU) thresholds ranging from 0.5 to 0.95, with a step size of 0.05. mAP\_s@0.5:0.95, mAP\_m@0.5:0.95, and mAP\_l@0.5:0.95 refer to small, medium, and large objects, respectively. The best results are marked in bold

Method	mAP@0.5:0.95	mAP@0.5	mAP@0.75	mAP_s@0.5:0.95	mAP_m@0.5:0.95	mAP_l@0.5:0.95	Memory cost(GB)
R50-FPN+L1Loss (our baseline)	0.404	0.616	0.472	0.101	0.557	0.485	4.66
R50-FPN+GiouLoss [37]	0.398	0.603	0.466	0.107	0.529	0.487	4.67
R50-FPN+CiouLoss [27]	0.428	0.633	<b>0.510</b>	0.128	0.576	0.520	4.67
R50-FPN+L1Loss+NMS [28]	0.422	0.627	0.490	0.100	0.561	0.509	4.67
R50-FPN+FP16+NMS+L1Loss	0.424	0.630	0.503	0.102	0.559	0.523	<b>3.29</b>
R50-FPN+FP16+NMS+GiouLoss	0.422	0.634	0.496	0.082	0.558	0.506	<b>3.29</b>
R50-FPN+FP16+NMS+CiouLoss	0.432	0.635	0.498	0.096	0.575	0.530	2.98
R50-FPN+FP16+NMS+CiouLoss+OHEM [30]	0.434	<b>0.639</b>	0.507	0.131	0.580	0.522	3.30
<b>R50-FPN+FP16+NMS+CiouLoss+OHEM+DST</b>	<b>0.458</b>	0.636	0.489	<b>0.192</b>	<b>0.591</b>	<b>0.531</b>	3.30

**Table 3** Performance comparison of different algorithms and models. “-” indicates that the data are not provided or that this metric is not used. The best results are marked in bold

Method	mAP@0.5:0.95	mAP@0.5	FPS_1	Memory cost(GB)	Multi-stage	FPS_2	FPS_3
Yolov3_mobilenetv2_416 [38]	0.239	–	14.70	5.30	No	23.80	24.10
Yolox_tiny [39]	0.320	–	8.70	<b>3.50</b>	No	15.40	15.90
Yolov3_d53 [40]	0.338	–	<b>16.20</b>	3.80	No	<b>33.10</b>	<b>34.30</b>
Retinanet_r50_fpn [41]	0.374	–	5.40	3.80	No	8.90	9.40
Faster R-CNN [17]	–	0.531	–	4.67	Yes	–	–
Improved Faster R-CNN [17]	–	0.574	–	–	Yes	–	–
Faster R-CNN+L1loss	0.404	0.616	4.70	4.00	Yes	9.10	10.50
FabricNet [15]	–	0.620	–	–	Yes	–	–
<b>Faster R-CNN+Fine-tuned (ours)</b>	<b>0.458</b>	<b>0.636</b>	4.90	4.67	Yes	9.30	10.20

mAP\_s@0.5:0.95 increase by 5.4%, 2%, and 9.1%, respectively.

As shown in Table 2, replacing L1Loss of the baseline with CiouLoss increases the mAP@0.5:0.95 by 2.4%. With NMS and FP16, the mAP@0.5:0.95 value improves by 2% compared to that of the baseline, and the GPU memory cost decreases by 1.37 GB, reducing the GPU workload. Compared to the optimized models with NMS and FP16, using OHEM and replacing L1Loss with CiouLoss increases the mAP@0.5:0.95 value by 1%. Using the DST as a training strategy and following the optimization methods above, the mAP@0.5:0.95 score improves by 2.4% compared to not using this method. Overall, the mAP@0.5:0.95 score increases by 5.4% compared to the baseline.

### 6.2.2 Ablation experiment

The performance of the proposed model is significantly enhanced when the FP16 + NMS + CiouLoss + OHEM + DST method is used simultaneously, as demonstrated in Table 3. In addition to evaluating the performance of the model on the dataset, we also include a live test session to determine the real-time frame rate for detection on the proposed system, including FPS\_1, FPS\_2 and FPS\_3. FPS\_1 refers to the frames per second (FPS) in the user interface of the IPC program when both the VSOIC and the

VSODD modules are used. FPS\_2 refers to the FPS when VSOIC is enabled but VSODD is disabled. FPS\_3 refers to the FPS when both VSOIC and VSODD are disabled.

Compared to the studies of Wang et al. [17] and Zhou et al. [15], the network and the strategy proposed in this work achieve better mAP values on the dataset. However, it is not possible to compare their detection speeds on the system proposed in this work because the model of the above work is not open source. In addition, two-stage networks such as the Faster R-CNN cannot compete with networks such as YOLO and Retinanet in terms of detection speed, real-time frame rate, and GPU memory cost (during deployment). In particular, some decision algorithms of the detection system are executed directly at the output of the model. Although this slows down the detection speed of the models, we still consider the introduction of the metric “FPS” (used in the proposed system, including FPS\_1, FPS\_2, and FPS\_3) in the model performance comparison phase to be significant and in line with the original intent of our work.

### 6.3 Function demonstration

According to the degree of defects, the fabric to be detected is divided into defect-free fabric, defective fabric, and fabric with large-area defects, and the corresponding

post-processing process is designed for three situations, as shown in Fig. 10. In addition, the optimization of the scheduling and the time consumption of each process are implemented, so that the system is more stable, and the operation logic is self-consistent. The time consumption and scheduling of the functional demonstrations are shown in Fig. 11.

### 6.3.1 Anomaly detection

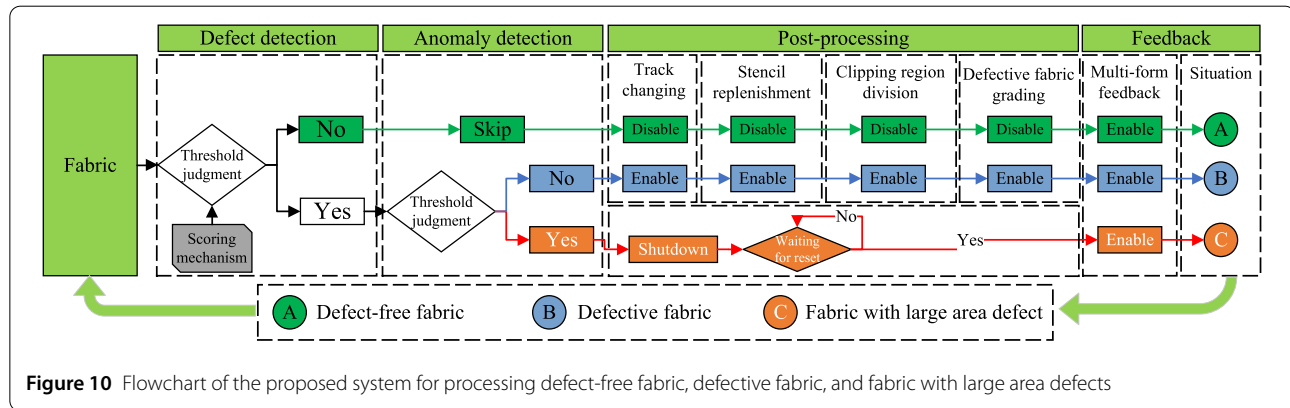
When a piece of fabric that exceeds the defect score alarm threshold is fed to the bottom conveyor, the IPC successfully sends an alarm command to the alarm light via the development board, causing all conveyors to shut down and display a warning tip in the user interface. When the operator clicks the “module disable” button on the large-area defect automatic shutdown module in the user interface, the module can be reset, and the entire system can resume its operation. Fig. 12(a) shows the operation and results of the system. As a result, this anomaly detection module can effectively solve the problem of large defects caused by ab-

normal conditions of a knitting machine by using vision technology-based automatic shutdown.

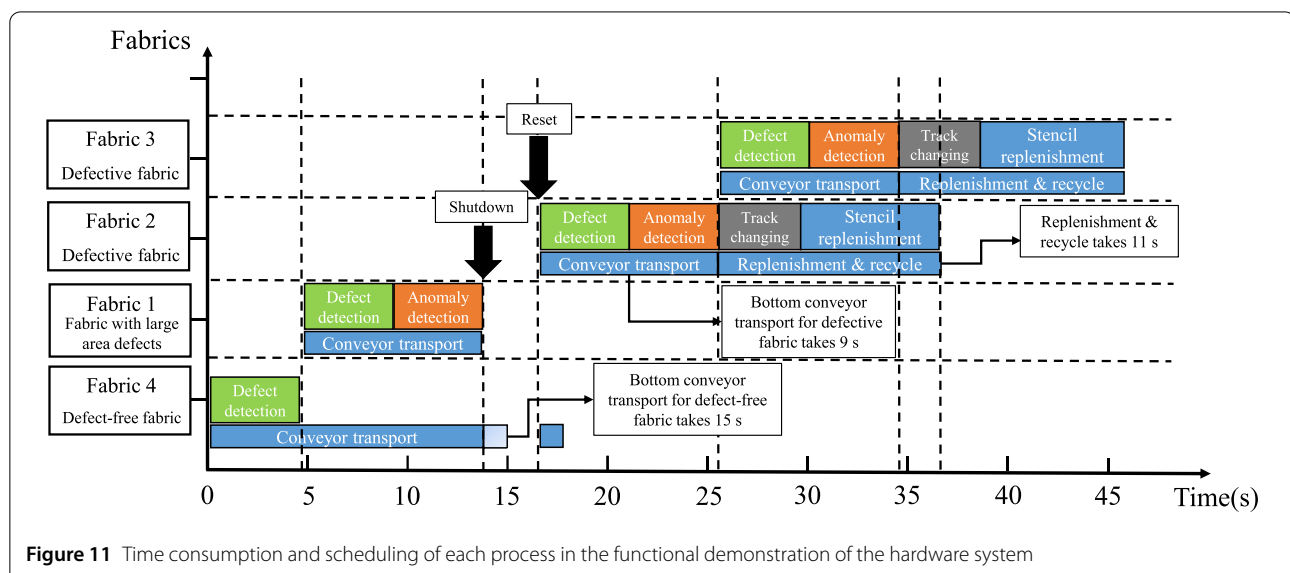
### 6.3.2 Defect detection and post-processing

The experimental results for defective fabrics with different stencils are shown in Fig. 12(b) and Fig. 12(c). During the defective fabric detection experiment, the user interface returns the current fabric score, and the system enters the defective fabric processing stage. At the same time, a release command is sent to the fabric replenishment module, and the pattern number and information are displayed on the user interface. The replenishment module replenishes the defect-free fabric recycle station with defect-free fabric in same stencil. Meanwhile, the defective fabric recycle conveyor transfers the defective fabric itself to the defective fabric replenishment station.

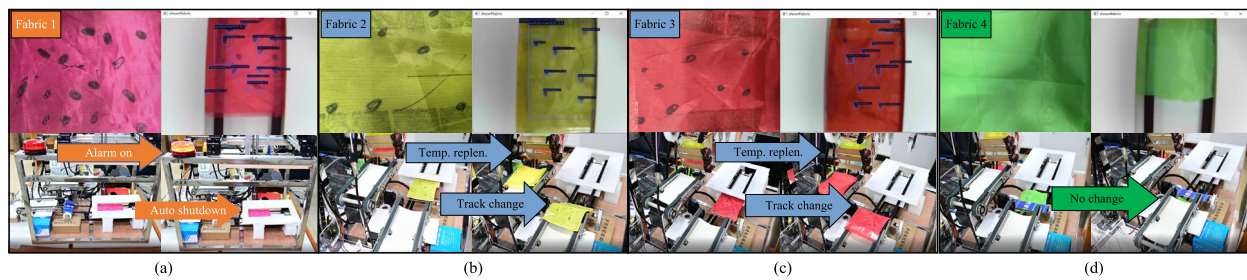
The results of a defect-free fabric delivery experiment are shown in Fig. 12(d). The user interface displays the current fabric defect score without activating the defect processing module, and the fabric continues to pass through



**Figure 10** Flowchart of the proposed system for processing defect-free fabric, defective fabric, and fabric with large area defects



**Figure 11** Time consumption and scheduling of each process in the functional demonstration of the hardware system



**Figure 12** Fabric image, defect identification, evaluation and processing on (a) shows the test of anomaly detection. There is no stencil identification, and the score is 265. (b) shows the test of the defective fabric for stencil 1. The result of the stencil identification is stencil 1 with a score of 150. (c) displays the test of the defective fabric for stencil 4. The result of the stencil identification is stencil 4 with a score of 125. (d) displays the test of defect-free fabric. There is no stencil identification, and the score is 10

the conveyor as normal. The fabric replenishment module does not receive an “enable” command. As a result, this system is shown to be capable of detecting and processing defective full-color fabrics.

### 6.3.3 Multi-form information feedback

The system also provides a review function for fabrics with detected defects, which enables the output of detection results in the form of video streams, screenshots, GIFs, and logs. In addition, clipping region division experiments and defective fabric grading experiments are performed on the defective fabrics tested in the above experiments.

## 7 Conclusion and future work

Due to complex scenarios and high market demands that drive continuous research efforts, fabric defect detection and post-processing offer promising computer vision applications. We propose a fabric defect detection and post-processing system that integrates an optimized Faster R-CNN model for defect detection, defect localization and detection model evaluation. In addition, the system incorporates new approaches, such as a rearranged fabric dataset, anomaly detection, recommended clipping region division, and a replenishment device. Meeting the requirements of practical industrial production can be more valuable than incremental improvement, so computer vision and automation technology have been combined here, to bridge scientific research and industrial production.

However, resolving the conflict between detecting small defects and large receptive fields is still a challenging task. The efficiency of real-time detection needs to be improved, since 5 ~ 10 frames per second is not enough for fabric manufacturing. In addition, the feedback of the results provided by the proposed system to the machine operators also needs to be improved. In future research, we will explore potential solutions for companies to address the above issues, improve the optimization of the system’s operational logic and scheduling scheme, and develop new

features such as defect tagging (which uses additional hardware modules to provide feedback at the physical level by attaching tags to defects). To encourage further research in this area, our approach is available as open source on GitHub at [https://github.com/linhuaizhou/yida\\_gedc\\_fabric4show](https://github.com/linhuaizhou/yida_gedc_fabric4show).

### Abbreviations

bbox, bounding box; CNN, convolutional neural networks; DST, double-stream training; FPS, frames per second; GPU, graphics processing unit; IPC, industrial personal computer; mAP, mean of average precision; OHEM, online hard example mining; VSODD, video stream output of defect detection; VSOIC, video stream output of industrial camera.

### Acknowledgements

The detection and post-processing system mentioned in this work received assistance from the Guangdong Esquel Textiles Co.,Ltd., and was awarded the first prize and the best paper prize in the 17th GEDC (Graduate Student Electronic Design) National Final Technical Competition.

### Author contributions

Huaizhou Lin conceived the idea of the study and wrote the paper. Zengmin Xu collected data and performed experiments. Dan Cai and Lixian Sun revised the basic idea in detail. Jinsong Wu and Haibin Jia contributed to the main idea and revision of the paper. All authors have read and approved the final manuscript.

### Authors’ information

The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request, and also are available from Aliyun Tianchi and Kaggle [22–25].

### Funding

This work was supported in part by the Free Exploration Fund of Guangxi Key Laboratory of Structure-Activity Relationship of Electronic Information Materials (No. 211019-Z), Guangxi Natural Science Foundation (No. 2024GXNSFAA010493), Innovation Project of GUET Graduate Education (No. 2022YCX5193), Chinesisch-Deutsche Kooperationsgruppe (No. GZ1528), National Key Research and Development Program (No. SQ2022YFB4000136), and Science and Technology Development Project of Guilin (No. 20210102-4).

## Declarations

### Competing interests

The authors declare no competing interests.

# Author details

<sup>1</sup>School of Materials Science and Engineering, Guangxi Key Laboratory of Information, Guilin University of Electronic Technology, Guilin, China. <sup>2</sup>School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Center for Applied Mathematics of Guangxi (GUET), Guilin, China. <sup>3</sup>Anview.ai, Guilin, China. <sup>4</sup>School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin, China. <sup>5</sup>Department of Electrical Engineering, University of Chile, Santiago, Chile.

Received: 30 June 2023 Revised: 3 May 2024 Accepted: 6 May 2024  
Published online: 03 June 2024

# References

- Wong, C. (2017). *Applications of computer vision in fashion and textiles*. Cambridge: Woodhead Publishing.
- Kumar, A. (2008). Computer-vision-based fabric defect detection: a survey. *IEEE Transactions on Industrial Electronics*, 55(1), 348–363.
- Kahraman, Y., & Durmuşoğlu, A. (2023). Deep learning-based fabric defect detection: a review. *Textile Research Journal*, 93(5–6), 1485–1503.
- Wang, P.-H., & Lin, C.-C. (2022). Data augmentation method for fabric defect detection. In *Proceedings of the IEEE international conference on consumer electronics* (pp. 255–256). Piscataway: IEEE.
- Wang, S., Lv, C., Wang, S., Zhang, Z., & Shang, X. (2021). Patterned fabric defect detection based on double-branch parallel improved faster R-CNN. In *Proceedings of the Chinese automation congress* (pp. 3798–3803). Piscataway: IEEE.
- Chen, M., Yu, L., Zhi, C., Sun, R., Zhu, S., Gao, Z., et al. (2022). Improved faster R-CNN for fabric defect detection based on Gabor filter with genetic algorithm optimization. *Computers in Industry*, 134, 103551.
- Fang, B., Long, X., Sun, F., Liu, H., Zhang, S., & Fang, C. (2022). Tactile-based fabric defect detection using convolutional neural network with attention mechanism. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–9.
- Zhang, J., Jing, J., Lu, P., & Song, S. (2022). Improved mobilenetv2-ssdlite for automatic fabric defect detection system based on cloud-edge computing. *IEEE Transactions on Instrumentation and Measurement*, 201, 111665.
- Zhang, H., Qiao, G., Lu, S., Yao, L., & Chen, X. (2023). Attention-based feature fusion generative adversarial network for yarn-dyed fabric defect detection. *Textile Research Journal*, 93(5–6), 1178–1195.
- Hanbay, K., Talu, M. F., & Özgüven, Ö. F. (2016). Fabric defect detection systems and methods—a systematic literature review. *Optik*, 127(24), 11960–11973.
- Karlekar, V. V., Biradar, M. S., & Bhargale, K. B. (2015). Fabric defect detection using wavelet filter. In *Proceedings of the international conference on computing communication control and automation* (pp. 712–715). Piscataway: IEEE.
- Li, N., Zhao, J., & Jiang, P. (2017). Fabric defects detection via visual attention mechanism. In *Proceedings of the Chinese automation congress* (pp. 2956–2960). Piscataway: IEEE.
- Liu, J., Gong Zhang, B., & Li, L. (2020). Defect detection of fabrics with generative adversarial network based flaws modeling. In *Proceedings of the Chinese automation congress* (pp. 3334–3338). Piscataway: IEEE.
- Zhang, H.-W., Zhang, L.-J., Li, P.-F., & Gu, d. (2018). Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks. In *Proceedings of the IEEE 7th data driven control and learning systems conference* (pp. 170–174). Piscataway: IEEE.
- Zhou, H., Jang, B., Chen, Y., & Troendle, D. (2020). Exploring faster R-CNN for fabric defect detection. In *Proceedings of the third international conference on artificial intelligence for industries* (pp. 52–55). Piscataway: IEEE.
- Khodier, M. M., Ahmed, S. M., & Sayed, M. S. (2022). Complex pattern jacquard fabrics defect detection using convolutional neural networks and multispectral imaging. *IEEE Access*, 10, 10653–10660.
- Wang, S., Lv, C., Wang, S., Zhang, Z., & Shang, X. (2021). Patterned fabric defect detection based on double-branch parallel improved faster R-CNN. In *Proceedings of the Chinese automation congress* (pp. 3798–3803). Piscataway: IEEE.
- Liu, J., Jiang, J., & Zhang, Q. (2017). Research on geo-relationship network and competing (or mutually beneficial) relationship network of Chinese textile enterprises. In *Proceedings of the 29th Chinese control and decision conference* (pp. 6462–6467). Piscataway: IEEE.
- Zhu, X., Liu, Z., Zhang, X., & Li, M. (2021). A visual distortion detection method for textile cloth. In *Proceedings of the 6th international symposium on computer and information processing technology* (pp. 113–116). Piscataway: IEEE.
- Konstantinidis, F. K., Mouroutsos, S. G., & Gasteratos, A. (2021). The role of machine vision in industry 4.0: an automotive manufacturing perspective. In *Proceedings of the IEEE international conference on imaging systems and techniques* (pp. 1–6). Piscataway: IEEE.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In *Proceedings of the 29th international conference on neural information processing systems* (pp. 91–99). Red Hook: Curran Associates.
- Textile defect detection dataset. Retrieved November 5, 2023, from: <https://tianchi.aliyun.com/dataset/94213>.
- Smart diagnosis of cloth flaw dataset. Retrieved November 5, 2023, from: <https://tianchi.aliyun.com/dataset/79336>.
- Bridge cracks image dataset. Retrieved November 5, 2023, from <https://www.kaggle.com/datasets/yidazhang07/bridge-cracks-image>.
- Fabric defect dataset. Retrieved November 5, 2023, from <https://www.kaggle.com/datasets/rmshashi/fabric-defect-dataset>.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). Piscataway: IEEE.
- Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU loss: faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 12993–13000). Palo Alto: AAAI Press.
- Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS—improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision* (pp. 5561–5569). Piscataway: IEEE.
- Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., et al. (2017). Mixed precision training. arXiv preprint [arXiv:1710.03740](https://arxiv.org/abs/1710.03740).
- Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 761–769). Piscataway: IEEE.
- Wang, T.-J., Peng, J.-Y., & Hung, Y.-F. (2016). Modeling fabric cutting scheduling as mixed integer programming. In *Proceedings of the IEEE international conference on industrial engineering and engineering management* (pp. 922–926). Piscataway: IEEE.
- Zhang, X., Lu, G., Cai, S., Hu, Z., Xiong, X., & Zhao, Q. (2018). Survey on intelligent leather nesting and cutting machine technology. In *Proceedings of the 11th international workshop on human friendly robotics* (pp. 84–88). Piscataway: IEEE.
- Phakphonhamin, V., & Chudokmai, M. (2018). Optimizing the performance of the lectra automatic fabric cutting machine. In *Proceedings of the 5th international conference on business and industrial research* (pp. 282–287). Piscataway: IEEE.
- Tiwari, V., Pandey, C., Dwivedi, A., & Yadav, V. (2020). Image classification using deep neural network. In *Proceedings of the 2nd international conference on advances in computing, communication control and networking* (pp. 730–733). Piscataway: IEEE.
- Barbu, T., Costin, M., & Ciobanu, A. (2009). Histogram intersection based image retrieval technique using relevance feedback. In *Proceedings of the 3rd international workshop on soft computing applications* (pp. 67–70). Piscataway: IEEE.
- Wei, W., Deng, D., Zeng, L., & Zhang, C. (2021). Real-time implementation of fabric defect detection based on variational automatic encoder with structure similarity. *Journal of Real-Time Image Processing*, 18(3), 807–823.
- Rezatofghi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: a metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 658–666). Piscataway: IEEE.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510–4520). Piscataway: IEEE.
- Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: exceeding YOLO series in 2021. arXiv preprint [arXiv:2107.08430](https://arxiv.org/abs/2107.08430).
- Redmon, J., & Farhadi, A. (2018). YoloV3: an incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767).



41. Du, X., Zoph, B., Hung, W.-C., & Lin, T.-Y. (2021). Simple training strategies and model scaling for object detection. arXiv preprint [arXiv:2107.00057](https://arxiv.org/abs/2107.00057).

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)