sensors

MDPI

Article

Online Detection of Fabric Defects Based on Improved CenterNet with Deformable Convolution

Jun Xiang, Ruru Pan and Weidong Gao *

School of Textile Science & Engineering, Jiangnan University, No. 1800, Lihu Avenue, Wuxi 214122, China; skyjun@163.com (J.X.); prrsw@163.com (R.P.)
* Correspondence: gaowd3@163.com

Abstract: The traditional manual defect detection method has low efficiency and is time-consuming and laborious. To address this issue, this paper proposed an automatic detection framework for fabric defect detection, which consists of a hardware system and detection algorithm. For the efficient and high-quality acquisition of fabric images, an image acquisition assembly equipped with three sets of lights sources, eight cameras, and a mirror was developed. The image acquisition speed of the developed device is up to 65 m per minute of fabric. This study treats the problem of fabric defect detection as an object detection task in machine vision. Considering the real-time and precision requirements of detection, we improved some components of CenterNet to achieve efficient fabric defect detection, including the introduction of deformable convolution to adapt to different defect shapes and the introduction of i-FPN to adapt to defects of different sizes. Ablation studies demonstrate the effectiveness of our proposed improvements. The comparative experimental results show that our method achieves a satisfactory balance of accuracy and speed, which demonstrate the superiority of the proposed method. The maximum detection speed of the developed system can reach 37.3 m per minute, which can meet the real-time requirements.

Keywords: fabric defect detection; feature pyramid network; deformable convolution; object detection; online detection

Citation: Jun, X.; Ruru, Pan.; Weidong, G. Online Detection of Fabric Defects Based on Improved CenterNet with Deformable Convolution. *Sensors* **2022**, *22*, 4718. https://doi.org/10.3390/s22134718

Academic Editor: Kelvin K.L. Wong

Received: 10 May 2022
Accepted: 6 June 2022
Published: 22 June 2022

1. Introduction

In the weaving process of fabrics, due to the influence of the technological process, weaving equipment, or weaving environment, it is inevitable to cause various defects on the surface of fabrics. The appearance of defects will not only affect the appearance of the fabric, but also reduce the commercial value of the fabric. Relevant reports [1] show that if there are obvious defects in the surface of the fabric, its price will be reduced by more than 50%; therefore, defect detection is an important step in fabric quality control; however, at present, most textile enterprises still rely on manual cloth inspection, which not only has the shortcomings of low efficiency and high cost, but is also prone to false detection or missed inspection after visual fatigue. With the advancement of digitization and intelligence, the development of fabric defect detection towards automation is an inevitable trend.

The automatic detection of fabric defects mainly includes two steps: firstly, images of the fabric surface are captured by using an industrial camera, and then the existence and type of defect in the image are judged by designing a recognition algorithm. The detection methods based on computer vision have the advantages of high precision, high efficiency, and strong stability; therefore, the automatic detection of fabric defects by machine vision instead of human vision has become a research hotspot; however, as shown in Figure 1, the main characteristics of the defects in the fabric are as follows: (1) rich types and different shapes and (2) low visual significance, which makes the identification task very challenging.

(**a**) (**b**) (**c**)

Figure 1. Three kinds of defects on the fabric surface. (**a**) Regional defects in fabrics; (**b**) Weft defect in fabric; (**c**) Warp defect in fabric

Efficient defect detection methods can greatly reduce labor consumption, so many methods have been proposed. The existing work on fabric defect detection can be roughly divided into four categories: (1) statistical-based, (2) spectral-based, (3) model-based, and (4) learning-based. The statistical methods [2,3] employ various statistical properties of texture and defects to estimate defects; however, the diversity of fabric texture and defect shape seriously affects the detection accuracy of such methods. In particular, it is very expensive to design different statistical indicators for defects of different complexity; therefore, statistical methods have great limitations in actual fabric defect detection. The spectral methods [4,5] convert the image in the spatial domain to the frequency domain and achieve the detection of defects in the fabric by using the strong periodicity in the fabric image; however, such methods do not work well when the contrast between defect areas and defect-free areas is low or when the defects are small. The model-based methods [6,7] represent fabric texture as a stochastic process and assume that texture images can be viewed as samples generated by stochastic processes in the image space. Defect detection is treated as a hypothesis testing problem with statistics from the model. Such methods usually have large computational overhead, and thus cannot meet the real-time requirements of detection; however, if a model-based algorithm is introduced into defect detection of fabric, a specific model for each texture is required, and the cost of each model is prohibitive.

Recently, significant progress [8–11] has been made on image analysis by moving low feature-based algorithms to deep-learning-based end-to-end frameworks. Compared with other kinds of methods, the deep-learning-based methods weaken the influence of feature engineering on recognition accuracy, adopt supervised or semi-supervised learning to make the network automatically extract the most representative features, simplify the design difficulty of the algorithm, automatically learn the salient features of the image, and complete the recognition task. Many researchers [12,13] use deep learning technology to solve the problem of fabric defect detection. Compared with earlier combined methods, deep-learning-based methods can extract higher-level features of images. According to the different learning manner, it can be divided into supervised learning and unsupervised learning. In the unsupervised manner, model learning is guided through designed pretasks. The general steps are: first reconstruct the fabric image, then compute the residual between the reconstructed image and the original image, and finally determine the location and category of the defect by identifying the residual image. Convolutional autoencoders (CAE) [14] and generative adversarial networks (GAN) [15] are the commonly used reconstruction models. Li et al. [16] first introduced deep learning technology into field of fabric defect detection by proposed an autoencoder model. Even with some success, such indirect methods are difficult to identify for many non-obvious defects.

In fact, fabric defect detection can be regarded as an object detection task, where the object is the defect. Compared with unsupervised methods, object detection can obtain more sufficient defect information, which is convenient for subsequent visual display and equality in judgment. Due to the different emphasis on detection speed and detection

accuracy, object detection methods are gradually developed in two directions: one stage and two stage. The two-stage methods, of which RCNN [17–19] is the most representative method, achieve high accuracy, but lose a certain detection speed. According to reports, the detection speed of Cascade RCNN [20] can only reach 14 fps, which cannot meet the real-time requirements of fabric defect detection. The classic one-step object detection methods are SSD [21] and YOLO [22]. Jing et al. [23] used the improved YOLOv3 to achieve efficient detection of six classical defects. The defect area in the fabric usually only occupies a small part, that is, the background area is much larger than the foreground area. This characteristic of fabric defects limits the performance of these methods. Recently Duan et al. [24] proposed a detector, named CenterNet, which detects each object as triplet keypoints, which can avoid the confusion brought by a large amount of background. CenterNet achieves a good trade-off between accuracy and speed, promising real-time defect detection.

Although deep-learning-based object detection methods have been partially studied in the industrial field, most of them are still in the laboratory stage and are difficult to implement for two reasons: (1) fabric defects are complex and diverse, making it difficult to detect and locate them in complex background areas; (2) online detection has high requirements for real-time performance, but most of the existing research ignores its speed; however, there is still potential for improvement when it is applied to fabric defect detection.

In this paper, we propose a fabric defect detection method based on CenterNet with deformable Convolution for online detection.

2. Theoretical Basis

2.1. Multi-Resolution CenterNet Module

CenterNet is an efficient bottom-up object detection method, which solves the problem that traditional methods have, i.e., they lack additional attention for proposed regions. The authors design a real-time version of CenterNet, whose framework is shown in Figure 2. In the framework of CenterNet, ResNet-50 [25] is used as backbone. The feature maps extracted by C3-C5 then connected to a FPN, to capture multi-scale feature of the input image. Then, the outputs P3-P5 of FPN are mapped as the final prediction layers. In each prediction layer, the regression is used to prediction the keypoints. In fact, the main innovation of CenterNet is a key point prediction network named KPN. The architecture of KPN is shown in Figure 3. As shown in Figure 2, KPN receives the output of FPN, and then outputs the predicted keypoints through some convolutional transformations. After obtaining the keypoints, the location of the bounding box can be determined. The learning of KPN is driven by the IoU loss, which is defined by:

$$\mathcal{L}_{IoU} = \frac{B_g \cap B_p}{B_g \cup B_p} - \frac{d^2(C_g, C_p)}{d_m^2} \tag{1}$$

where B_g and B_p denote the ground truth and predicted bounding box, respectively; C_g and C_p are the center points of B_g and B_p; $d(\cdot, \cdot)$ represents the Euclidean distance of the two points; d_m represents the diagonal distance of the smallest closure region that can contain both the predicted and ground-truth boxes. In regression-based regression, to decouple the top-left and the bottom-right corners, the ground truth box is divided into four sub-ground truth boxes along the geometric center. Among the sub-ground truth boxes, the top-left and bottom-right are selected to supervised the regression, respectively. During the inference, the regressed vectors act as a cue to find the nearest keypoints on the corresponding heatmaps to refine the locations of the keypoints. Next, each valid pair of keypoints defines a boundary box. Finally, a central region is defined for each bounding box and check if the central region contains both the predicted center keypoints. If there is at most one center keypoint detected in its central region, the bounding box will be removed. The score of the bounding box will be replaced by average scores of the points.

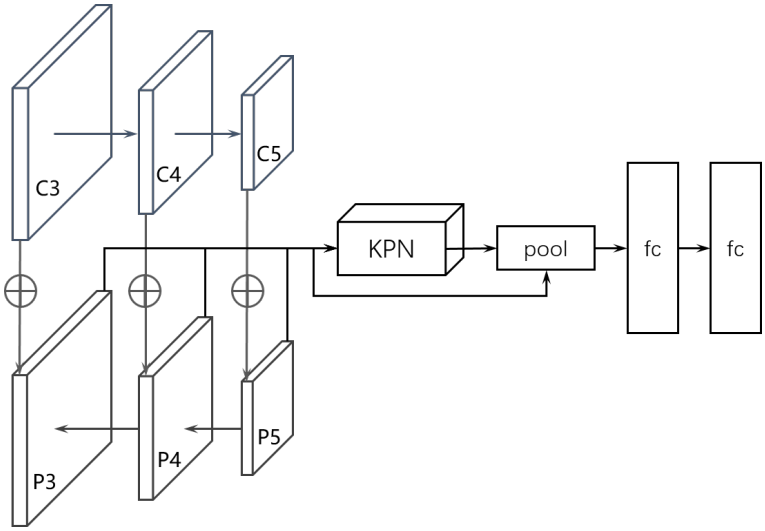


Figure 2. Real-time detection framework of CenterNet. The backbone outputs three feature maps, which are C3-C5, to connect a feature pyramid network (FPN). Then FPN outputs P3-P5 feature maps as the final prediction layers.

CenterNet draws on the residual structure to extract deep feature information and multi-scale feature to improve the performance of different scale objects. While improving the detection performance, the used explicit FPN tends to obtain limited receptive field. Simply increasing the number of block will result in large parameter burden and memory consumption. So CenterNet still has some room for improvement in speed.

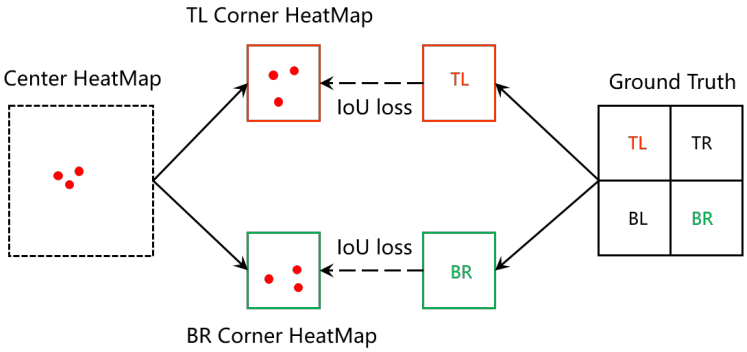


Figure 3. The architecture of key prediction network (KPN). TL is top-left corner, TR is top-right corner, BL is bottom-left corner, BR is bottom-right corner, and IoU is intersection over union.

2.2. Deformable Convolution Module

In recent years, with the popularity of deep convolutional neural networks, many difficult vision problems have achieved major breakthroughs. Image recognition [9] first surpassed human recognition abilities more than two years ago. The accuracy of object detection [17–19,26], image segmentation, etc., has also reached a height that is difficult to achieve by traditional methods. Due to the powerful modeling ability and automatic end-to-end learning method, deep convolutional neural networks can learn effective features from a large amount of data, avoiding the drawbacks of artificially designed features in traditional methods; however, the adaptability of existing network models to the geometric deformation of objects almost entirely comes from the diversity of the data itself, and the model does not have a mechanism to adapt to the geometric deformation. The fundamental reason is that the convolution operation itself has a fixed geometric structure, and the geometric structure of the convolutional network built by its stacking is also fixed, so it does not have the ability to model geometric deformation. Tracing the source, the above limitations come from the basic building block of the convolutional networks—the

convolution operation. This operation performs sampling based on the regular grid point position at each position of the input image, and then convolves the sampled image value as the output of that position. Through end-to-end gradient back-propagation learning, the system will obtain a matrix of convolution kernel weights. This is the basic unit structure that has been used for more than two decades since the birth of convolutional networks. Researchers at Microsoft Research Asia found that regular lattice sampling in standard convolutions is the "culprit" that makes the network difficult to adapt to geometric deformations [27]. To weaken this limitation, the researchers added an offset variable to the location of each sampling point in the convolution kernel. Through these variables, the convolution kernel can be randomly sampled near the current position, instead of being limited to the previous regular grid points. This expanded convolution operation is called deformable convolution, as shown in Figure 4.

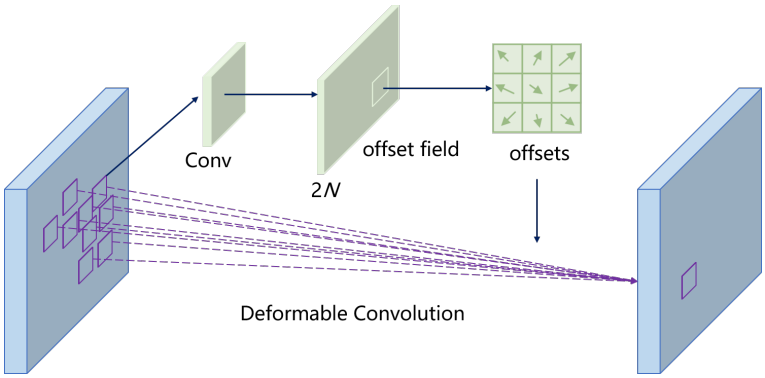


Figure 4. Illustration of 3 × 3 deformable convolution [27].

The 2D convolution consists of two steps: (1) sampling using a regular grid $\mathcal{A}$ over the input feature map $\boldsymbol{x}$; (2) summation of sampled values weighted by $\boldsymbol{w}$. The grid $\mathcal{A}$ defines the receptive field size and dilation. For example,

$$\mathcal{A} = \{(-1,-1),(-1,0),...,(0,1),(1,1)\} \tag{2}$$

defines a 3 × 3 kernel with dilation 1. For each location p_0 on the output feature map y, we have

$$\boldsymbol{y}(p_0) = \sum_{p_n \in \mathcal{A}} \boldsymbol{w}(p_n) \cdot \boldsymbol{x}(p_0 + p_n) \tag{3}$$

where p_n enumerates the locations in $\mathcal{A}$. In deformable convolution, the regular grid $\mathcal{A}$ is augmented with offsets $\{\triangle\ p_n | n = 1, ...N\}$, where $N = |\mathcal{A}|$. Then, the above equation can be rewritten as:

$$\boldsymbol{y}(p_0) = \sum_{p_n \in \mathcal{A}} w(p_n) \cdot \boldsymbol{x}(p_0 + p_n + \triangle\ p_n) \tag{4}$$

Thus the sampling is on the irregular and offset locations $p_n + \triangle\ p_n$. As the offset $\triangle\ p_n$ is typically fractional, Equation (4) is implemented via bilinear interpolation as

$$\boldsymbol{x}(p) = \sum_q \mathcal{B}(p,q) \cdot \boldsymbol{x}(q) \tag{5}$$

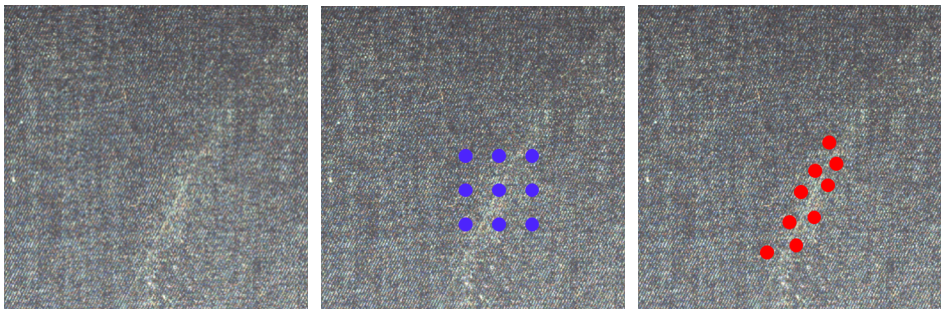
where p denotes an arbitrary (fractional) location ($p = p_0 + p_n + \triangle\ p_n$ for Equation (4)), q enumerates all integral spatial locations in the feature map $\boldsymbol{x}$, and $\mathcal{B}(\cdot,\cdot)$ is the bilinear interpolation kernel with 2D; it is separated into two one-dimensional kernels as

$$\mathcal{B}(p,q) = \beta(q_x, p_x) \cdot \beta(q_y, p_y) \tag{6}$$

where $\beta(a,b) = \max(0, 1 - |a - b|)$. Then, $\mathcal{B}(p,q)$ can be computed quickly.

In fact, the added offset in the deformable convolution unit is part of the network structure, calculated by another parallel standard convolution unit, which in turn can also be learned end-to-end by gradient back-propagation. After adding this offset learning, the size and position of the deformable convolution kernel can be dynamically adjusted according to the current image content to be recognized. The intuitive effect is that the positions of the convolution kernel sampling points at different positions will adaptively change according to the image content, so as to adapt to the geometric deformations, such as the shape and size of different objects.

As shown in Figure 5, the shape of defects in the fabric is irregular, so this paper proposes to use deformed convolution to adapt to the shape of different defects. Using this type of convolution allows the model to more precisely locate the defective area, and thus more accurately identify the defect type.

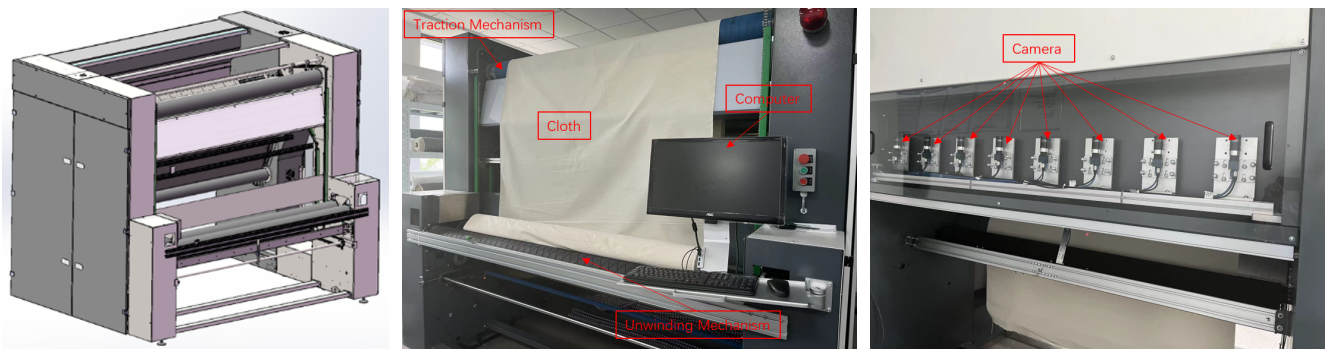


(**a**) (**b**) (**c**)

Figure 5. Different convolutions. (**a**) Defective image; (**b**) traditional convolution with the kernel size of 3×3; (**c**) deformable convolution with the kernel size of 3×3 .

3. Hardware System

In this section, the key components in hardware system are introduced in detail. Figure 6 shows the overall diagram of the developed equipment, which consist of an unwinding mechanism, traction mechanism, winding mechanism, image acquisition component, and computer. The frequency conversion motor realizes the unwinding, pulling and winding of the cloth by controlling the rotation of the roller. When the cloth passes through the image acquisition area, the camera automatically captures the fabric image and sends it to the software system in the computer for detection, as shown in Figure 7. Apart from the image acquisition component, the developed equipment is similar to other automatic defect inspection equipment; therefore, this section focuses on the introduction of image acquisition component.



(**a**) (**b**) (**c**)

Figure 6. Hardware system. (**a**) Overall diagram of the equipment; (**b**) front view of the equipment; (**c**) rear view of the equipment.

In real-time inspection, the choice of camera is an important factor to obtain high-quality fabric images. There are two types of industrial cameras commonly used in defect detection: line-scan cameras and area-scan cameras. This paper studies defect detection technology on the basis of surface images, so the area-scan camera was selected as the image acquisition device. In the developed equipment, eight industrial cameras (MER-502-79U3M) were arranged linearly, which can realize the rapid acquisition of fabric images. To ensure stability, a lighting system with three light sources and a reflector was designed.

In practical applications, the size of the fabric image captured by each camera is 2430 × 1200 pixel, which corresponds to the actual size of the fabric is 29.0 × 14.3 cm (84 pixels/cm, 0.119 mm/pixel). The width of the overlapping area between the images captured by adjacent cameras is about 1.8 cm. The equipment can realize defect detection of fabrics with a maximum width of 2.2 m. If the detection time is ignored, the developed device can achieve image acquisition of 65 m of cloth per minute.

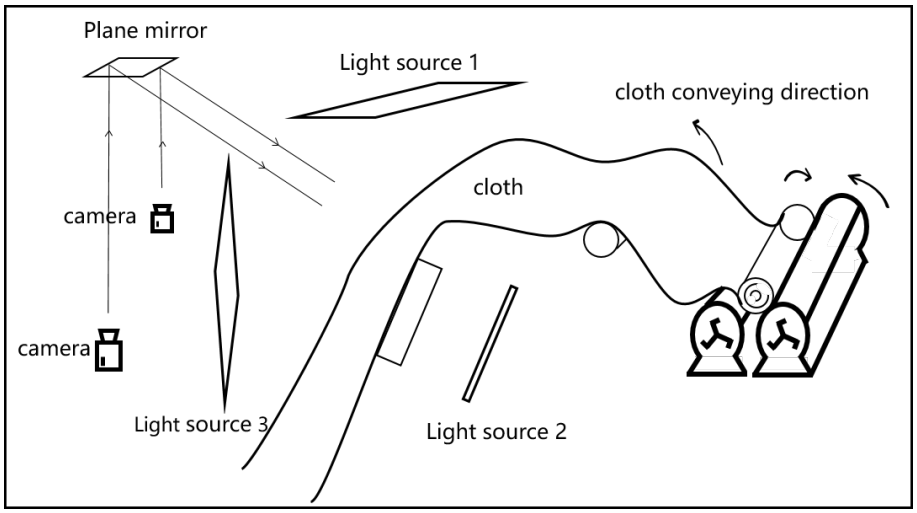


Figure 7. The internal structure diagram of the developed automatic cloth inspection equipment.

4. Detection Algorithm

In this section, we introduce the proposed detection algorithm in detail, and the network architecture is presented in Figure 8. Similar to the original CenterNet, we still use ResNet50 [25] as the backbone, but some of the convolutional layers are replaced by deformable convolutions. Secondly, an implicit feature pyramid network (i-FPN) is introduced for two purposes: (1) to enhance the detection performance of the model for small defects; (2) to speed up the detection. Then, we introduce the objective function of the improved CenterNet. Finally, an online detection framework for fabric defects is built using the trained model. It is stated here that Figures 2 and 8 are not the same, we replace the original explicit FPN with i-FPN.

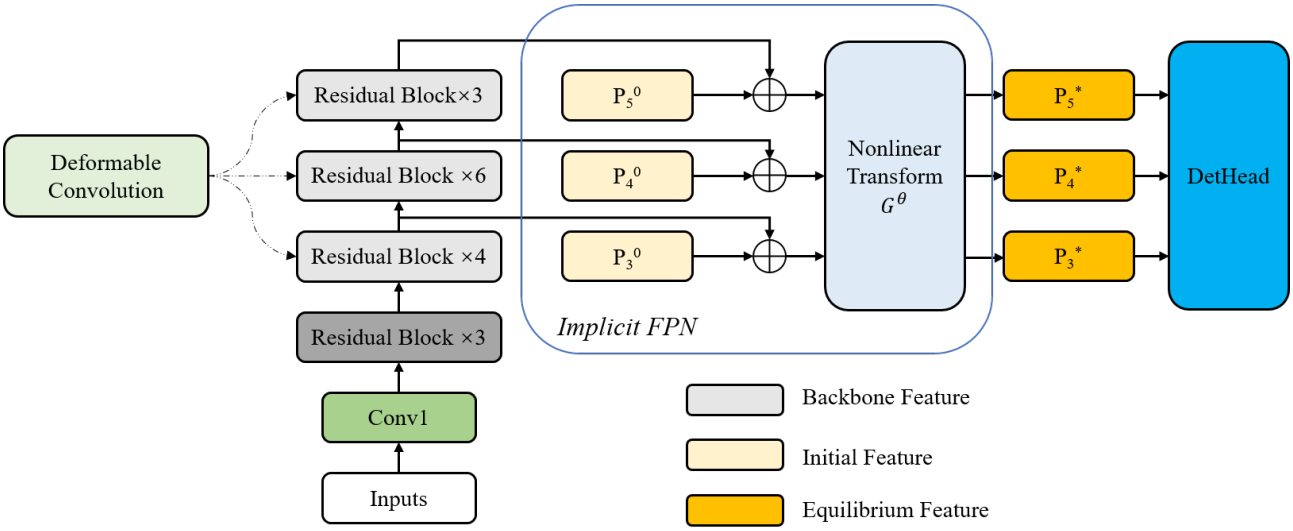


Figure 8. Architecture of object detector with implicit feature pyramid network. The ResNet50 [25] is adopted as the backbone network to extract backbone features. The initial pyramid features, which are all initialized to zeros, together with the backbone features are input to the i-FPN. In the i-FPN, the nonlinear transformation function G^θ is employed to construct the implicit function and the equilibrium feature pyramid is injected into detection head to generate the final detection predictions.

4.1. Backbone Network

Although defects of various shapes and sizes only destroy the original texture structure of the fabric, the task of defect detection is a highly abstract task to a certain extent, because many defects are not the most prominent in the fabric image. In general, the deeper the convolutional neural network can extract, the more abstract features present; however, increasing the network depth brings some problems: (1) difficulty of convergence and (2) overfitting. ResNet introduces a residual structure into the network model, making it possible for the network depth to exceed 100 layers. The introduced residual makes it easier for the network to learn the identity mapping at some layers, which is a constructive solution. Residual networks behave similar to an ensemble of relatively shallow networks. In addition, the residual network allows information to flow between layers, and features can be reused during forward propagation, which alleviates the risk of gradient disappearance or gradient explosion during back propagation. In summary, ResNet can extract more abstract features without overfitting. ResNet50 and ResNet101 are two architectures that are often used as backbones; however, considering the real-time requirements of defect detection, we chose the former as the backbone of the proposed detection model.

To accommodate defects of different shapes, we introduce deformable convolutions in the backbone network. The idea of deformable convolution is very simple, that is, the original fixed-shape convolution kernel becomes variable. Taking the 3×3 convolution kernel as an example, the mathematical expression is as follows:

$$\mathbf{y}(\mathbf{p_0}) = \sum_{\mathbf{p_n} \in \mathcal{R}} \mathbf{w}(\mathbf{p_n}) \cdot \mathbf{x}(\mathbf{p_0} + \mathbf{p_n}) \tag{7}$$

where $\mathcal{R}$ represents the set of points in the neighborhood of $\mathbf{p_0}$, and $\mathbf{n}$ is the index of the point in the $\mathcal{R}$. For the output $\mathbf{y}(\mathbf{p_0})$ of each convolution, it needs to sample from nine positions on the feature map $\mathbf{x}$, of which, nine positions are determined by the center position $\mathbf{p_0}$. The deformable convolution operation does not change the calculation operation of the convolution, but adds a learnable parameter $\nabla \mathbf{p_n}$ to the convolution region. Similarly, for each output $\mathbf{y}(\mathbf{p_0})$, nine positions must be sampled from the input feature map. These nine positions are obtained by diffusing the center position $\mathbf{p_0}$ to the surroundings,

but with more $\nabla \mathbf{p_n}$, the sampling points are allowed to spread into a non-grid shape. The deformable convolution operation can be expressed as:

$$\mathbf{y}(\mathbf{p_0}) = \sum_{\mathbf{p_n} \in \mathcal{R}} \mathbf{w}(\mathbf{p_n}) \cdot \mathbf{x}(\mathbf{p_0} + \mathbf{p_n} + \nabla \mathbf{p_n}) \tag{8}$$

To learn the offset $\nabla \mathbf{p_n}$, another 3×3 convolutional layer needs to be defined. In fact, as shown in Figure 4, the size of the output offset field is the same as that of the original feature map, but the number of channels is twice the original (representing the offset in the x and y directions, respectively). In this case, with the input feature map and the offset field of the same size as the feature map, we can perform deformable convolution operations. The above operations are all differentiable processes, so the parameters can be learned through backpropagation.

To combine the advantages of ResNet and deformable convolution, we improve some residual blocks of ResNet50. As shown in Figure 8, the improvement is mainly reflected in the latter three series of residual blocks. Specifically, as shown in Figure 9, for each residual structure, we use a 3×3 deformable convolution to replace the original 3×3 ordinary convolution; the other architectures are exactly the same as the original ResNet50—we refer the interested reader to [25].

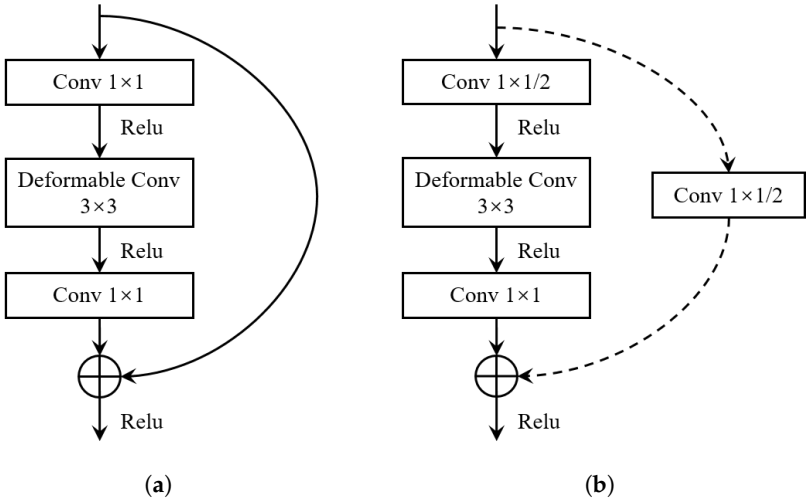


Figure 9. Two residual blocks (three layers) with deformable convolutions. (**a**) The bottleneck layer that makes the shape of the feature map invariant; (**b**) the bottleneck layer that reduces the length and width of the feature map to half.

4.2. Implicit Feature Pyramid Network

To enhance the performance of the detector for objects of different scales, the commonly used method is explicit feature pyramid network (FPN), which stacks several cross-scale blocks to obtain large receptive field. It has been proved that implicit FPN (i-FPN) has better performance than explicit FPN, mainly in terms of detection speed and robustness. Different from explicit FPN, i-FPN directly produces equilibrium feature of global receptive field based on fixed point iteration. In addition, a recurrent mechanism, named residual-like iteration, is introduced to efficiently update the hidden states for feature pyramid design.

The architecture of i-FPN can be seen in Figure 8. i-FPN generates an equilibrium feature pyramid based on fixed point iteration. The initial features P_3^0–P_5^0 are all initialized to zeros. It is then fed into the i-FPN along with the backbone feature. The summed feature is input into the nonlinear transformation G^θ, which serves as the implicit function. The equilibrium feature solver is further employed to generate the equilibrium feature pyramid by solving the fixed point of the implicit model. Finally, the resulting equilibrium

feature pyramids are injected into the detection head to generate the final classification and regression predictions.

Figure 10 presents the explicit form of i-FPN, which is named residual-like iteration, to simulate explicit FPN with infinite depth. The residual-like iteration can be formulated as:

$$\mathrm{P}^* = G^{\theta}(\mathrm{P}^* + \mathrm{B}) \tag{9}$$

where P^* can be computed by the unrolling solver or Broyden solver in DEQ [28]. Similar to ResNet [25], the residual-like iteration can also benefits from the residual learning by shortcut connection. The backbone features, which are extracted by backbone network and served as the strong prior, guide the residual learning of nonlinear transformation G^{θ}, as shown in Figure 11; therefore, the residual-like iteration can prevent i-FPN from suffering from the vanishing gradient problem, and theoretically, an FPN of infinite depth can be obtained. The ingenious structure of iFPN enables smooth information propagation, which enhances feature learning. Consequently, the equilibrium feature pyramid is input into detection head to recognize the keypoints, bounding boxes, and classes.

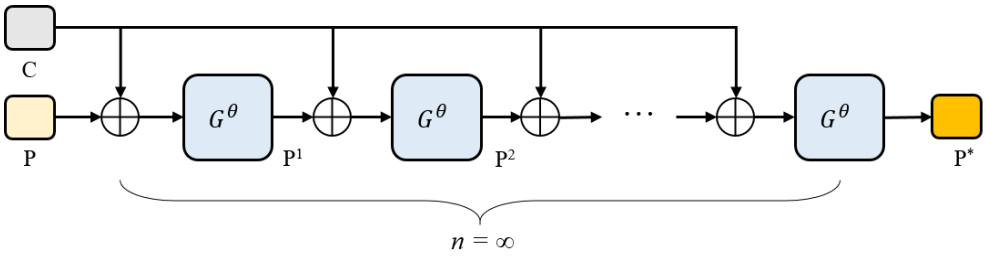


Figure 10. The pipeline of residual-like iteration. Note that $n = 3$ in this paper.

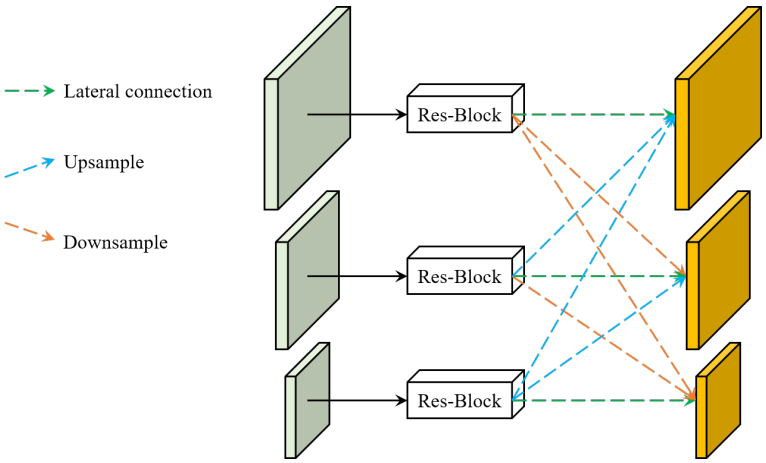


Figure 11. The architecture of the nonlinear transformation G^{θ}. The dash lines denotes pyramid convolution to reduce the computation redundancy and efficiently fuse cross-scale features.

4.3. Detection Head

As shown in Figure 3, the keypoints serve as the basic object representation throughout CenterNet. The keypoints are obtained via regressing offsets over the center points, which are predicted by KPN (mentioned in Section 2.1). The learning of the keyponts are driven by two loss function: the bottom-right and top-left IoU loss between the induced pseudo box and the ground truth bounding box; the object recognition loss of the subsequent stage. The architecture of the detection head is illustrated in Figure 12. The proposed head architecture consists of two non-shared subnets, aiming at localization and classification, respectively. The localization subnet first uses three 3×3 convolutional layers, followed by two consecutive small networks to compute the offsets of the two sets of keypoints. The classification subnet also uses three 3×3 convolutional layers to abstract the feature maps,

followed by a deformable convolutional layer whose input offset field is shared with the first deformable convolutional layer in the localization subnet. The group normalization layer is applied after each of the first three 3×3 convolutional layers in the two subnets. The anchor-free design reduces the burden on the final classification layer, resulting in a slight reduction in computation.

As shown in Figure 12, localization subnet consist of two stages: generating the first set of keypoints by abstraction from object center point hypotheses (feature map bins); generating the second set of keypoints based on the first set of keypoints. During training, only positive target hypotheses are assigned to localize targets for both stages. For the first localization stage, there are two conditions for a feature map bin to be considered positive: (1) the pyramid level of this feature map bin is equal to the logarithmic scale of the real object; (2) the projection of the center point of this real object is located in this feature map bin. For the second localization stage, it is positive if the induced pseudo-box of the first keypoints have enough overlap with a real object, and their intersection over-union is greater than 0.5. Classification is only conducted on the first set of keypoints. The classification assignment criteria follow: IoU (between the induced pseudo-box and the ground-truth bounding box) greater than 0.5 means positive, less than 0.4 means background, otherwise ignored. Focal loss [29] is used for classification task training.

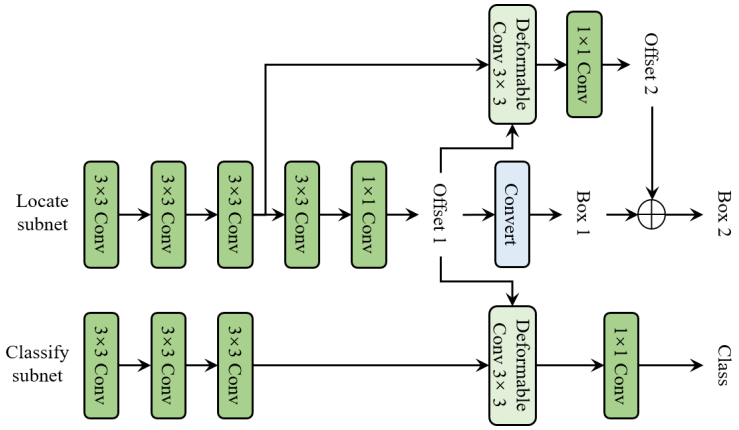


Figure 12. The pipeline of detection head.

5. Experiment

5.1. Experimental Dataset

As we all know, the defect detection method based on deep learning learns the defect localization and recognition ability from a certain amount of training data; therefore, data are the basis for model learning. To train the model and verify the effectiveness of the method, we use the public fabric defect dataset (Smart Diagnosis of Cloth Flaw Dataset, SDCFD) [30], in which the samples are all from the production line of the textile factory. SDCFD contains 11,918 fabric RGB images, of which 2842 are used as a testing set to test the method performance and 9076 are used as a training set to train the model. There are 5913 defect images in the training set, which cover 34 defect types. The size of the images in this dataset is 2446 pixel $\times$ 1000 pixel. For defect detection, SDCFD provides bounding box annotations which are saved as an json document, indicating the category and the location of defect in each image. To facilitate the analysis, the fabric defects are visually divided into three categories: warp defects (length-width ratio less than 0.5), weft defects (length-width ratio greater than 2), and regional defects (otherwise).

The size of the fabric image collected by the proposed equipment in this paper is 2430×1200 pixels, which is similar to the image resolution of SDCFD, and the shooting scale is basically the same; therefore, the model trained on this dataset can be directly grafted onto the equipment for online detection.

5.2. Evaluation Criteria

Different from the classification task, the fabric defect detection not only needs to predict the correct category but also the location information of the defect. In this study, we use three types of indicators to evaluate the performance of the defect detection methods from different perspectives; we also use three types of metrics to evaluate the performance of the defect detection methods from different perspectives. The recall R, detection rate D_R, false detection rate F_R, and detection accuracy D_{ACC} are used to evaluate the recognition performance of the detection method; the mean average precision mAP is used to evaluate the localization performance of the detection method; the FPS (frames per second) is used to evaluate the time complexity of the method.

R and D_R measure the ability of the model detection for positives, D_{ACC} measures the accuracy of the model prediction, and F_R reflects the robustness of the model. The three metrics are computed as follows:

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$D_R = \frac{TP}{N_{\text{defect}}} \tag{11}$$

$$F_R = \frac{FP}{N_{\text{defect-free}}} \tag{12}$$

$$D_{ACC} = \frac{TP + TN}{FP + FN + TN + TP} \tag{13}$$

where N_{defect} and $N_{\text{defect-free}}$, respectively, denote the total number of detective and defect-free images. The definitions of TP, FN, FP, and TN are presented in Table 1.

Table 1. Definition of TP, FN, FP, and TN in fabric defect detection.

	Detected as Defective	**Detected as Defect-Free**
Actually defective	True Positive (TP)	False Negative (FN)
Actually defect-free	False Positive (FP)	True Negative (TN)

AP is the area under the P-R curve corresponding to a certain category of detection results, and mAP is the average value of the area under the P-R curve corresponding to the detection results of all categories. In this study, we calculate the AP based on 11-point interpolation method, which can be defined as:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} \rho_{\text{interp}(r)} \tag{14}$$

where

$$\rho_{\text{interp}(r)} = \max_{\tilde{r} \geq \tilde{r}} \rho(\tilde{r}) \tag{15}$$

where $\rho(\tilde{r})$ is the measured precision at recall $\tilde{r}$. When AP for classes are obtained, the mAP can be computed by:

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{16}$$

where K represents the number of classes.

FPS represents the number of images that can be recognized per second, which is used to measure the time complexity of the detection algorithm. It is stated here that smaller FR values indicate better model performance, while the values of other metrics are positively correlated with method performance.

5.3. Implementation Details

The appearance of defects in solid-colored fabrics generally destroys the original texture characteristics of the fabrics; therefore, defects can be visually identified only from grayscale images. To meet the real-time requirements of defect detection, this paper proposes to grayscale the RGB image first, and then input the model for training or testing.

Compared to large-scale datasets, such as COCO [31], the SDCFD used in this paper are relatively small. Under such conditions, data augmentation is an effective means to enhance the recognition accuracy and generalization of the model. During the training process, we randomly perform some transformations on the input fabric image, including grayscale transformation, rotation transformation, flip transformation, cropping, affine transformation, and so on. In terms of parameter setting, the input size is 1333 $\times$ 800 pixel, the initial learning rate is 5×10^{-3}, weight decay is 5×10^{-4} and the total epoch is 50. To avoid training falling into local optimum, at the 30th and 40th epoch, the learning rate is adjusted to $\frac{1}{10}$ of the previous epoch.

In this study, the proposed method is implemented by using the Pytorch toolkit 1.9.0 + CUDA11.4 + cuDNN8.2.1. The hardware environment is as follows: CPU = E5 2623V4@ 2.60 GHz, RAM = DDR4 32G, and GPU = GeForce RTX 3090(24 G) $\times$ 2. Partial of visual results of detection on SDCFD-testing dataset are shown in Figure 13.

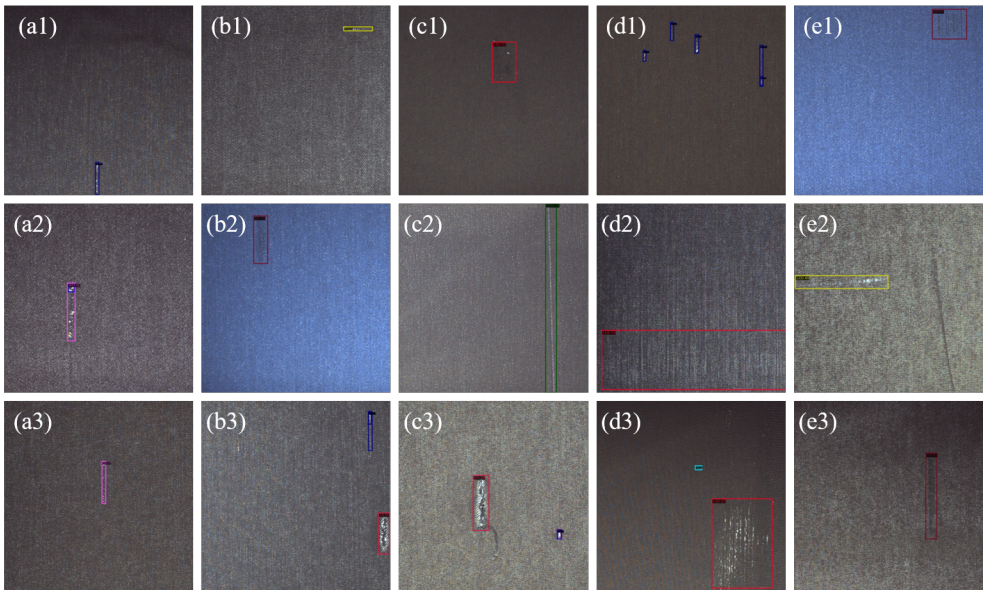


Figure 13. A partial visual result of the fabric defect detection. In each fabric image, the color boxes are the predicted bounding boxes. The color of the box represents the defect of the specified category. Among them, sub-images (a1, d1, a2, b2, c2, a3, b3, c3, e3) are warp defects; sub-images (b1, d2, e2) are weft defects; sub-images (c1, e1, d3) are regional defects.

5.4. Ablation Study

To validate the efficacy and efficiency of the proposed approach, we conduct a thorough ablation study in this subsection. Compared with the original CenterNet, our main improvements are as follows: (1) Deformable convolution is introduced to improve the adaptability to defects of various shapes; (2) FPN is replaced with i-FPN to improve the accuracy of small targets. All ablation experiments are conducted with ResNet50 backbone and evaluated on SDCFD-testing dataset.

We first explore the effect of using two different convolutions, namely common convolution and deformable convolution. Table 2 presents the performance comparison results. It is stated here that "Common convolution" in the table indicates that all the convolutions in the model are common convolutions, and "Deformable convolution" indicates that the partial convolutions (mentioned in the previous section) in the model are deformable convolutions. The baseline is "Common convolution", producing 0.527 box mAP. From the

results, it can be found that the model has a higher recognition rate for regional defects, but lower for warp and weft defects. It has been demonstrated that deformable convolution has strong detection performance for irregular objects. Moreover, most fabric defects are often irregular in shape. The model with deformable convolution achieves a average mAP of 0.648 with $+0.121$ improvement. Except mAP, other detection performance indicators for all categories have been improved to a certain extent, which proves the rationality and effectiveness of using deformable convolution instead of common convolution.

As mentioned before, i-FPN is another key component used to improve the recognition accuracy of the model for small defects. Here, we conduct the comparative experiment on SDCFD to analyze the effect of it, and define the defects that occupy an area less than 300 (the number of pixels in the area) in the original image as small defects. Table 3 and Figure 14 present the quantitative comparison results when adopting different FPN architectures as the cross-scale connection. The baseline is "None" (the first row in the Table 3) without the cross-scale connection. It is clearly observed that the detection performance of the model is significantly improved when cross-scale connection is adopted, especially for small defects. For example, comparing "None" and "FPN", D_R achieves an improvement of 0.102 for small defects. In addition, adopting Bi-FPN [32] or NAS-FPN [33] as cross-scale connection produces a decent performance with the mAP score of 0.531 and 0.548 while Dense-FPN provides more improvements. Moreover, as shown in Figure 14, i-FPN has great advantages in the detection performance of each category of defects. Further, i-FPN achieves more improvements on all evaluation criteria; therefore, using iFPN as the cross-scale connection can effectively improve the detection performance of the model for various defects, especially small defects.

Table 2. Performance comparison of the proposed model using common convolutions and deformable convolutions.

Configurations	Type	R	D_R	F_R	D_{ACC}	mAP
Common convolution	warp defects	0.818	0.827	0.108	0.854	0.481
	weft defects	0.809	0.823	0.112	0.848	0.469
	regional defects	0.847	0.859	0.057	0.882	0.586
	average	0.825	0.842	0.092	0.869	0.527
Deformable convolution	warp defects	0.876	0.924	0.051	0.927	0.624
	weft defects	0.881	0.931	0.047	0.924	0.623
	regional defects	0.960	0.983	0.017	0.958	0.752
	average	0.894	0.938	0.043	0.942	0.648

Table 3. Performance comparison between different design choices of cross-scale connection, include None, Bi-FPN, NAS-FPN, Dense-FPN, and i-FPN on SDCFD.

Types	Performance for Small Defects					Average				
	R	D_R	F_R	D_{ACC}	mAP	R	D_R	F_R	D_{ACC}	mAP
None	0.737	0.732	0.177	0.769	0.437	0.759	0.764	0.164	0.783	0.477
FPN [34]	0.796	0.834	0.104	0.842	0.517	0.828	0.826	0.095	0.834	0.529
Bi-FPN [32]	0.818	0.829	0.091	0.859	0.531	0.831	0.837	0.081	0.852	0.546
NAS-FPN [33]	0.825	0.864	0.084	0.868	0.548	0.839	0.875	0.076	0.879	0.568
Dense-FPN[35]	0.841	0.873	0.072	0.880	0.562	0.868	0.889	0.062	0.902	0.593
i-FPN	0.875	0.915	0.057	0.926	0.614	0.894	0.938	0.043	0.942	0.648

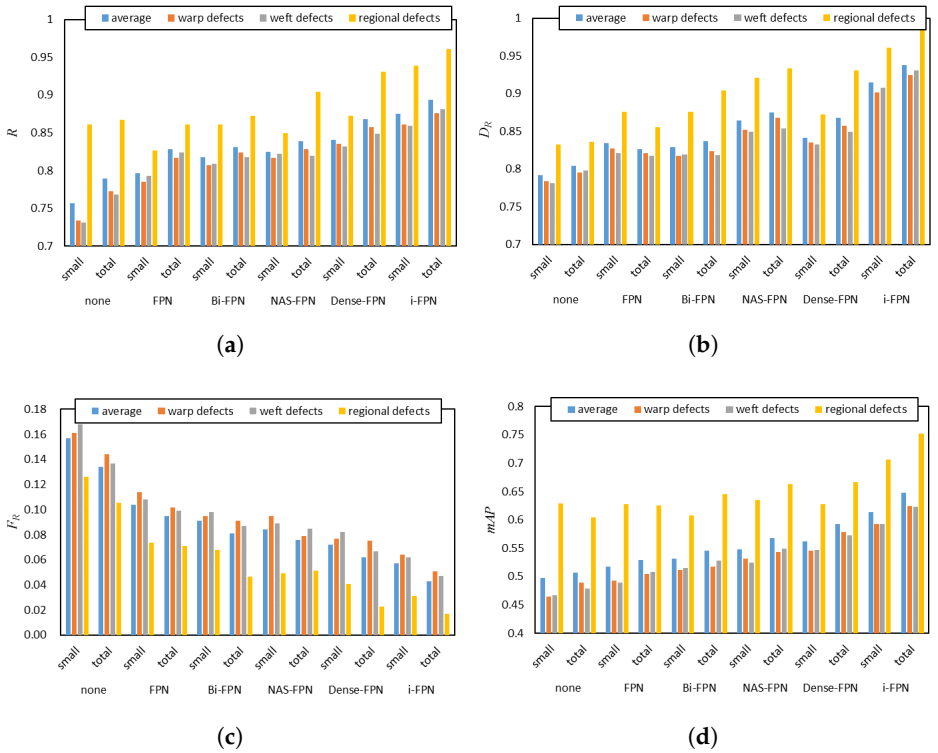


Figure 14. Performance comparison between different design choices of cross-scale connection on different type of defects. (**a**) Comparison of recall of different methods; (**b**) Comparison of Detection Rate of different methods; (**c**) Comparison of False-alarm Rate of different methods; (**d**) Comparison of mAP of different methods

To verify the superiority of the proposed method for fabric defect detection, we compare it with 10 other classical object detection methods, including one two-stage method: Faster R-CNN [19]; one multi-stage method: Cascade R-CNN [20]; two transformer-based methods: DETR [36] and Deformable DETR [37]; seven one-stage methods: YOLOv3 [38], SSD [21], CornerNet (anchor-free method) [39], M2det [40], RetinaNet [29], CenterNet-RT (anchor-free method), [24] and FCOS (anchor-free method) [41]. The performance comparison results are reported in Table 4.

Table 4. Comparison of the speed and accuracy of different object detector on SDCFD. We compare the results with batch = 1 without using tensorRT.

Methods	Backbone	R	D_R	F_R	D_{ACC}	mAP	FPS
Faster R-CNN [19]	ResNet50	0.806	0.816	0.128	0.825	0.427	13.5
Cascade R-CNN [20]	ResNet50	0.872	0.863	0.095	0.893	0.528	11.8
DETR [36]	ResNet50	0.859	0.861	0.098	0.860	0.492	10.8
Deformable DETR [37]	ResNet50	0.882	0.898	0.069	0.896	0.535	11.3
YOLOv3 [38]	DarkNet53	0.763	0.782	0.168	0.776	0.358	45.0
SSD [21]	VGG16	0.718	0.721	0.218	0.729	0.309	43.0
CornerNet [39]	Hourglass	0.749	0.763	0.231	0.752	0.349	6.5
M2det [40]	VGG16	0.763	0.775	0.184	0.769	0.319	33.4
RetinaNet [29]	ResNet50	0.792	0.785	0.163	0.791	0.315	16.2
CenterNet-RT [24]	ResNet50	0.858	0.875	0.073	0.862	0.593	30.5
FCOS [41]	ResNet50	0.834	0.847	0.105	0.851	0.549	26.1
Proposed	ResNet50	0.894	0.938	0.043	0.942	0.648	34.8

5.5. Comparisons

Regardless of the detection speed, Faster R-CNN and Cascade R-CNN must be the best choices for fabric defect detection. As shown in the first two rows of Table 4, these two methods achieve certain advantages in detection accuracy; however, their FPS indicators only reach 13.5 and 11.8, which cannot meet the real-time requirements of defect detection. DETR and Deformable DETR are all based on the transformer architecture [42], which is greatly affected by the size of the training data and thus achieve limited performance. As classic one-stage object detectors, YOLOv2 and SSD have great advantages in detection speed, which can detect 43 and 45 images per second, respectively; however, the performance achieved by these two detectors is not ideal in terms of accuracy, mainly due to their limited detection capability for small defects. Moreover, their false detection rate F_R is relatively high, which cannot be tolerated by textile enterprises. Although the three anchor-free methods CornerNet, CenterNet-RT, and FCOS have certain advantages in terms of computational complexity, their performance cannot meet the needs of defect detection. It is clear that the proposed method outperforms other methods for fabric defect detection, in terms of all evaluation criteria. The proposed method can detect 34.8 images per second, and when this model is grafted onto the proposed online detection device, the maximum detection speed can reach $34.8 \times 14.3 \times 60 \div 8 \div 100 = 37.3$ m/min. The average speed of manual cloth inspection is only 30m/min. In summary, the comparison results demonstrate that our methods achieves the best performance in all indicators, which proves the superiority of our proposed method. Combined with the proposed detection algorithm and the developed equipment, the detection speed can reach 37.3 m/min, which can meet the real-time requirements of defect detection.

5.6. Error Detection Analysis

By analyzing the samples of false detections, it is found that false detections mainly include over detection and missing detection. In Figure 15, we present some examples of false detections. The proposed method detects defects based on key points, and repeated detection may occur for independent defects that are close to each other, as shown in Figure 15(r1,g1); however, this false detection generally does not affect the final result of detection. Wrinkles and imperfections in fabrics are visually very similar and can therefore cause false detections, which are difficult to avoid, as shown in Figure 15(r2,g2). In addition, some defects only have a small number in the training set, making it difficult for the model to locate and identify them, as shown in Figure 15(r3,g3,r4,g4,r5,g5); however, we believe that when there are enough training samples in the training set, the proposed model can be sufficiently trained to accurately identify such defects.

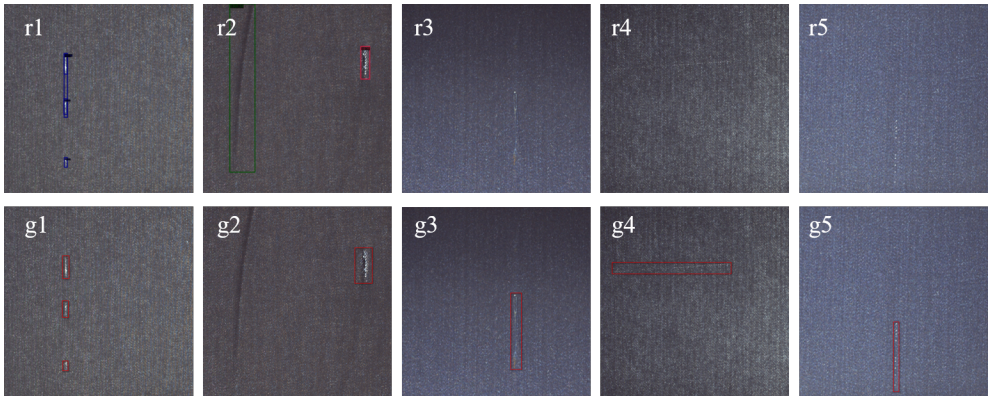


Figure 15. Some examples of false detections, where the first row shows the detection results and the second row shows the ground truth. Overdetection occurs in r1, g1, r2 and g2; and missed detection occurs in r3, g3, r4, g4, r5, g5.

6. Conclusions

In this paper, a novel automatic detection system for fabric defects was developed, which includes hardware system and detection algorithm. In the hardware system, three light sources and one mirror are configured to achieve efficient and high-quality acquisition of fabric images. This study defines the task of fabric defect detection as an object detection problem. Considering the real-time and accuracy requirements, we propose a defect detection method based on the improved CenterNet. Defects in fabric images generally have the characteristics of various shapes and sizes, so we introduce deformable convolution and i-FPN in CenterNet. Ablation experiments demonstrate that the two components can effectively improve the detection performance. Compared with other object detectors, the proposed method achieves the best performance in all indicators, which proves the superiority of proposed method. Moreover, compared with proposed detection method, the maximum detection speed of the developed equipment can reach 37.3 m/min, which can meet the real-time requirement of fabric defect detection.

Author Contributions: Conceptualization, R.P.; methodology, J.X.; software, J.X.; validation, J.X. and R.P.; formal analysis, J.X.; investigation, J.X.; resources, W.G.; data curation, J.X.; writing—original draft preparation, J.X.; writing—review and editing, R.P.; visualization, J.X.; supervision, W.G.; project administration, W.G.; funding acquisition, W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61976105.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The publicly archived fabric defect dataset (Smart Diagnosis of Cloth Flaw Dataset, SDCFD) can be downloaded at the following link: https://tianchi.aliyun.com/dataset/dataDetail?dataId=79336 (accessed on 21 October 2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiang, J.; Wang J.A.; Zhou, J.; Meng, S.; Pan, R.R.; Gao, W.D. Fabric defect detection based on a deep convolutional neural network using a two-stage strategy. *Text. Res. J.* **2021**, *91*, 130–142.
2. Bu, H.G.; Wang, J.; Huang, X.B. Fabric defect detection based on multiple fractal features and support vector data description. *Eng. Appl. Artif. Intell.* **2009**, *22*, 224–235. [CrossRef]
3. Mak, K.; Peng, P.; Lau, H. Optimal morphological filter design for fabric defect detection. In Proceedings of the 2005 IEEE international conference on industrial technology, Hong Kong, China, 14–17 December 2005; pp. 799–804.
4. Jia, L.; Chen, C.; Hou, Z. Fabric defect inspection based on lattice segmentation and Gabor filtering. *Neurocomputing* **2017**, *238*, 84–102. [CrossRef]
5. Bodnarova, A.; Bennamoun, M.; Latham, S. Optimal Gabor filters for textile flaw detection. *Pattern Recognit.* **2002**, *35*, 2973–2991. [CrossRef]
6. Yang, X.B. Fabric defect detection of statistic aberration feature based on GMRF model. *J. Text. Res.* **2013**, *34*, 137–142.
7. Allili, M.S.; Baaziz, N.; Mejri, M. Texture modeling using contourlets and finite mixtures of generalized Gaussian distributions and applications. *IEEE Trans. Multimed.* **2014**, *16*, 772–784. [CrossRef]
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceeding of the Advances in neural information processing systems, Lake Tahoe, NV, USA, 3–6 December 2012.
9. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Hassabis, D. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
10. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
11. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl. Sci.* **2018**, *8*, 1575. [CrossRef]
12. Liu, J.; Wang, C.; Su, H.; Du, B.; Tao, D. Multistage GAN for fabric defect detection. *IEEE Trans. Image Process.* **2019** *29*, 3388–3400. [CrossRef]
13. Jing, J.F.; Ma, H.; Zhang, H.H. Automatic fabric defect detection using a deep convolutional neural network. *Color. Technol.* **2019**, *135*, 213–223. [CrossRef]

14. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
15. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
16. Li, Y.; Zhao, W.; Pan, J. Deformable patterned fabric defect detection with fisher criterion-based deep learning. *IEEE Trans. Autom. Sci. Eng.* **2019**, *14*, 1256–1264. [CrossRef]
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 20–23 June 2014; pp. 580–587.
18. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceeding of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 11–12 December 2015.
20. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37
22. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
23. Jing, J.; Zhuo, D.; Zhang, H.; Liang, Y.; Zheng, M. Fabric defect detection using the improved YOLOv3 model. *J. Eng. Fibers Fabr.* **2020** *15*, 1558925020908268. [CrossRef]
24. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet++ for Object Detection. *arXiv* **2022**, arXiv:2204.08394.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
26. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2961–2969.
27. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 764–773.
28. Bai, S.; Kolter, J.Z.; Koltun, V. Deep equilibrium models. Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 10–12 December 2019.
29. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
30. Tian, C. Smart Diagnosis of Cloth Flaw Dataset. 2020. Available online: https://tianchi.aliyun.com/dataset/dataDetail?dataId=79336 (accessed on 21 October 2020).
31. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
32. Tan, M.; Pang, R.; Le Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, Aug 23–28. 2020; pp. 10781–10790.
33. Ghiasi, G.; Lin, T.Y.; Le Q.V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019; pp. 7036–7045.
34. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
35. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, YXiao, B. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020** *43*, 3349–3364. [CrossRef] [PubMed]
36. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
37. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 4 May 2021.
38. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
39. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 Sepember 2018; pp. 734–750.
40. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. M2det: A single-shot object detector based on multi-level feature pyramid network. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9259–9266.
41. Tian, Z.; Shen, C.; Chen, H.; F.C.O.S.; T.H. Fully Convolutional One-Stage Object Detection. In Proceedings of CVF International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 9626–9635.
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, California, OH, USA, 8 December 2017.