

Research Article

Fabric Defect Segmentation System Based on a Lightweight GAN for Industrial Internet of Things

Bo Li ¹, Yongkai Zou,¹ Rongbo Zhu ², Wei Yao,¹ Jun Wang,³ and Shaohua Wan⁴

¹College of Computer Science, South-Central Minzu University, Wuhan 430074, China

²College of Informatics, Huazhong Agricultural University, Wuhan 430070, China

³Yuxiang Technology (Hangzhou) Co. Ltd., Hangzhou 310024, China

⁴School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430074, China

Correspondence should be addressed to Rongbo Zhu; rbzhu@mail.hzau.edu.cn

Received 23 February 2022; Accepted 20 April 2022; Published 23 May 2022

Academic Editor: Varun Menon

Copyright © 2022 Bo Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine vision systems based on deep learning play an important role in the industrial Internet of things (IIoT) and Industry 4.0 applications, especially for product quality monitoring. Fabric defect detection is an important task in the industrial production of textiles and is crucial for product quality assurance. In actual production, the detection of many small and weak target defects remains challenging. Furthermore, industrial production requires high production rates and small model sizes in practice. This study proposes a lightweight segmentation system that meets real-time industrial production requirements. Herein, first, the defect sample image was repaired based on the image repair mechanism of the generative adversarial network model. Then, the difference between the defect sample and the repaired sample was obtained and subsequent processing, such as denoising and enhancement, was done. Finally, the defect areas were segmented. Our model was specifically designed for the segmentation of weak and small defects. This was achieved through adversarial training, optimization of an objective function, and image processing. Experimental comparisons show that the intersection over union of the three different datasets is 77.84%, 77.85%, and 73.6% and that our model is superior to the conventional semantic segmentation model. Furthermore, our model has good image restoration quality with a low mean absolute error and high structural similarity index. Additionally, our model is lightweight, has good real-time performance, and is suitable for applications in the IIoT and industrial production lines, such as embedded systems.

1. Introduction

In recent years, the industrial Internet of things (IIoT) has accelerated its integration into traditional industries and, therefore, has evolved into various applications. With the deployment of machine vision systems on the edge side, an automated production inspection line can be established for product defect detection; the inspection results can be transmitted to the cloud to provide data support to satisfy different customer needs. Migrating complete or partial tasks to the edge can diminish the network bandwidth, computing, and storage requirements of a cloud center [1]. Fabric defect detection has attracted significant research attention in the textile industry. In industrial production, it is essential to segment fabric defects to ensure the high qual-

ity of fabric products [2]. With the development of machine learning and machine vision technology, machine vision-based methods to solve textile quality control problems have gradually become an industry trend because of their high accuracy, fast detection speed, and low labor cost [3, 4].

Many researchers have used various algorithms and models for the automatic detection and segmentation of fabric defects [5, 6]. Two methods are applied for defect detection, one of which involves the use of classical image analysis algorithms, such as texture models [7], Fourier analysis [8], and Gabor filters [9].

The second method is based on deep learning algorithms [10, 11] that can often achieve good results. However, in practical applications, some problems remain, for example, compared to normal samples, fewer defect samples can be

obtained during production, and there are few observable types of defects. Additionally, the conventional labeling of defect samples is time-consuming and labor-intensive.

Traditional segmentation networks have insufficient segmentation capabilities for small and weak defect samples. In recent years, the generative adversarial network (GAN) model [12] has become increasingly favored and valued by researchers because of the strong modeling ability of the discriminator. It can continuously judge the difference between the segmentation results from the generator and the ground truth. The discriminator and the generator are optimized to obtain a segmentation feature map of multicontext features, which enables the generation of segmented images that are infinitely close to the ground truth. Therefore, in this study, we introduce a GAN as a fabric defect segmentation model.

Although the number of defect samples in the production is generally small and the types of defects that appear are also few, expanding the training sample set by heavy manual annotation work is not the best choice. By learning a small quantity of samples, the probability distribution of the texture and other features of the normal sample can be obtained. Some unknown random defects that are excluded from existing training samples often appear in industrial production. Regardless of the type and characteristics of the defect, differences exist between the normal sample area and the defect area. Therefore, the defect area can be determined based on these differences. Zhao et al. proposed a defect detection model based on positive samples, which first repairs the defect area and then determines the defect area by comparison [13]. The method of combining GANs and autoencoders is used to repair the defect image; then, local binary pattern (LBP) features are used to detect defects. The LBP method has a good effect on large-scale defects. However, the contours detected by the LBP features may be inaccurate for small and weak defects. In this study, we achieved the segmentation of the defect area based on the GAN image repair mechanism. In addition, many of the models have a large size and do not consider actual production needs.

Our research motivation is to develop a lightweight real-time system suitable for industrial production, confronting the more difficult detection problems of weak and small defects. Many existing models have good effects on conventional fabric defects. However, there are some small defects with low contrast, which affect the further improvement of product quality. Therefore, the detection and segmentation of these weak and small defects have become important research tasks. In addition, in actual production, the model must be as light as possible, occupy a small space, and meet real-time requirements. Deep learning systems are deployed at the edge and can play a very important role in the IIoT [14], and our system can be deployed on local production lines or at the edge of the IIoT.

In response to the abovementioned problems, herein, we studied the detection and segmentation of multiple types of defects in actual production samples, focusing on the segmentation of weak and small defects. A method for quickly constructing a large number of samples that conform to the true probability distribution is proposed.

In addition, a model designed and optimized to occupy a small space and have a fast segmentation speed is presented, which can also be applied to industrial fabric production. A GAN model was used to realize the segmentation of fabric defects. Adversarial training not only makes the model more stable but also increases the accuracy of defect segmentation.

In summary, the main contributions of this article are summarized as follows.

- (1) A fabric defect segmentation system suitable for industrial applications is proposed. The system is composed of a defect sample-synthesizing module without manual annotation, defect repair module, and defect segmentation module.
- (2) Using a combined image processing method, we designed a defect segmentation model for weak and small defects, including an objective function for confrontation training, a normalization method, and a learning rate decay strategy, which contribute to the accurate segmentation of defects.
- (3) The segmentation model proposed herein has the advantages of being lightweight and functions in real time, which is especially suitable for applications in IIoT and industrial production lines, such as embedded systems.

The remainder of this paper is organized as follows: Section 2 summarizes related work in recent years, Section 3 introduces our methodology, and Section 4 discusses the training process and model optimization method. Experimental results are presented and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Many classical algorithm models have been used in fabric defect segmentation and detection research, for example, wavelet analysis [15, 16], fuzzy C-means method [17], Gabor filter [18], established texture distribution model [19], Elo Rating algorithm [20], Bayesian classifier based on statistical features [21], and XGBoost classifier based on the genetic algorithm [22]. These methods have achieved good results in solving many application problems in related scenarios.

In recent years, with the development of deep learning theory, many valuable models have been proposed for semantic segmentation, such as the FCN [23], faster R-CNN [24], spatial pyramid pooling [25], U-Net [26], YOLO [27], RefineNet [28], SegNet [29], DeepLab [30], Fisher criterion [31], encoder-decoder [32], two-parallel-branch deep network [33], LSTM [34], EfficientDet [35], and multiscale network [36].

In the application field, fabric defect detection is also a type of object surface detection. Some researchers have conducted research in this field. For instance, surface defect detection is based on deep learning methods [37, 38].

In fabric production, defects are only a small part of the abnormal samples. In recent years, some researchers have

proposed anomaly detection methods and models [39] for image anomaly data detection. With the development of the GAN model, new models and theories have emerged consistently, such as conditional GANs [40, 41], cycle-consistent GANs [42], and style-based GANs [43]. In recent years, owing to the powerful learning ability of the GAN model, some researchers have also used GAN models for anomaly detection [44].

Schlegl et al. proposed AnoGAN [45], which is trained on positive samples to learn a mapping from the latent space. Akcay et al. subsequently proposed GANomaly [46]; their approach only requires a generator and a discriminator as in a standard GAN architecture, which is an improvement compared to AnoGAN and EGBAD. Perera et al. proposed OCGAN [47], and Ngo et al. proposed Fence-GAN, which corrects the GAN loss and has a better anomaly classification accuracy [48]. Zhao et al. proposed a defect detection framework based only on positive sample training [13]. The defect area in the sample is first repaired, and then, the model compares the input defect sample with the restored sample to determine the exact defect area. Furthermore, Wang et al. proposed a method using the GAN model with locality-preferred recoding for visual anomaly detection [49]. Nema et al. proposed an unpaired GAN model for brain tumor segmentation [50]. To identify small changes in small structures, Murugesan et al. proposed a new context-based loss function and a new architecture, Seg-GLGAN [51]. Liu et al. proposed a multistage GAN [52] model for fabric defect detection, which can automatically generate multiple defect samples.

In addition, Huang et al. recently adopted a deep learning model to segment defects, requiring only a small number of training samples [53].

In recent years, some researchers have proposed lightweight systems for applications, such as in underwater object [54], salient object [55], and blind road detection and crosswalks [56].

Although previous studies have made their own contributions, limited research on the balance of requirements of lightweight, real-time, and high-accuracy fabric defect detection and segmentation has been conducted in enterprise production. This article designs a system that includes three modules for the abovementioned problems and a model for the segmentation of weak and small defects.

3. Our Proposed Methodology

3.1. Problem Statement and Framework Description

3.1.1. Problem Statement. A fabric image with defects can be divided into normal and several defective regions. The normal region meets certain characteristics such as grayscale, color, and texture, whereas the defective regions do not meet the characteristics of the normal region.

Let I denote a fabric image and S denote a predicate with the same properties, and the image includes n regions R_i ($i = 1, 2, \dots, n$). Among them, R_1 is a normal region and the other regions are defective regions that satisfy

$$\begin{aligned} \bigcup_{i=1}^n R_i &= I, \\ R_i \cap R_j &= \emptyset, \\ S(R_i) &= \text{true}, \\ S(R_i \cup R_j) &= \text{false}. \end{aligned} \quad (1)$$

Defect segmentation results can be described as

$$I_{\text{mask}} = \begin{cases} 0, & \text{if } P(x, y) \in R_1, \\ 1, & \text{else,} \end{cases} \quad (2)$$

where $i = 1, 2, \dots, n, j = 1, 2, \dots, n$, and $i \neq j$. $P(x, y)$ is the image pixel, I_{mask} is the result after defect segmentation, and the region with a pixel value of one in I_{mask} is the defect region.

3.1.2. Framework Description. This study is a two-step segmentation method based on a GAN model, as shown in Figure 1. The first step is to repair the defect image to obtain the corresponding repaired image. The second step is to compare the two images to obtain the difference result and obtain the mask result of the defect area using image processing methods such as denoising, linear transformation, and binarization processing. The experiment was divided into three modules: a module that synthesizes defect samples, defect repair module, and defect segmentation module.

3.2. Synthesizing Defect Samples. Our experiments used samples taken from the equipment during the fabric production process. Owing to the rapid improvement of production processes, few defect samples can be obtained. To meet the needs of sample training, we designed a method to quickly obtain a large number of experimental samples without manual labeling. As shown in Figure 2, first, the defect areas are separated to obtain the defect block using only a small number of existing defect samples combined with the corresponding labeling information.

We used the “sliding cutting” method with a sliding step and cutting resolution. By sliding and cropping each sample, new images with cutting resolution were obtained. By determining whether the label corresponding to the cropped image contains a label, determining whether the cropped image is a defect image can be easy. In this manner, we can obtain a large number of normal samples with a cutting resolution.

Then, these few defect blocks are randomly pasted into the existing normal background by programming while recording the labeling information at the same time. In this way, a set of samples, including the defect, repaired, and mask images, is quickly obtained. Here, “random” includes the random selection of defect blocks and random pasting positions. Figure 2 shows an example of a method for artificially synthesizing and constructing defect samples.

This method not only quickly provides a large number of defect samples that are close to the original defect sample distribution but also directly obtains the corresponding

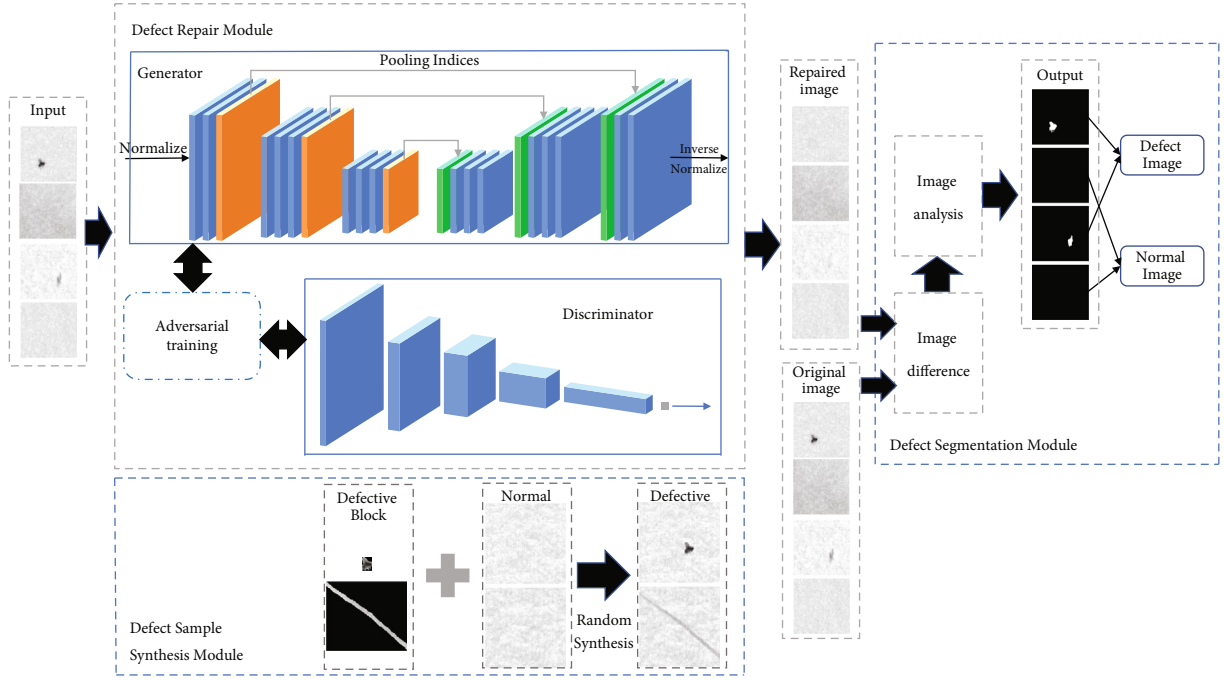


FIGURE 1: Architecture of our proposed system.

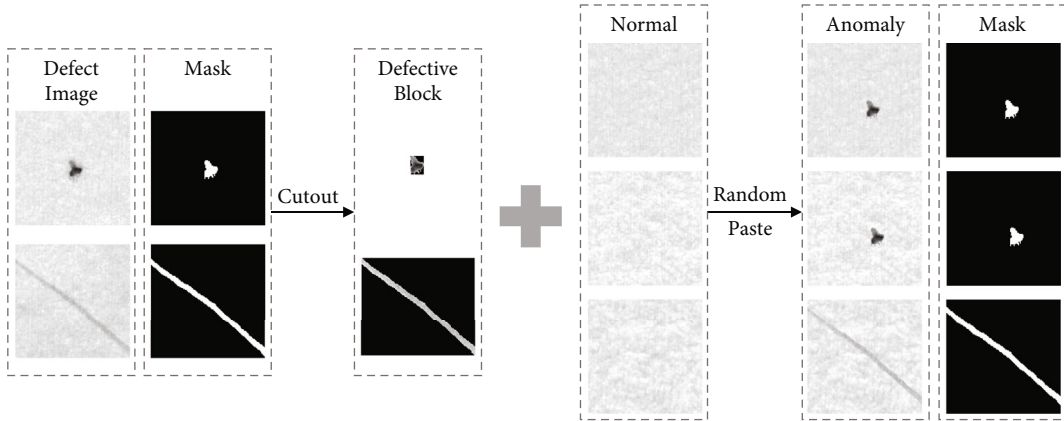


FIGURE 2: Method for synthesizing defect samples.

annotation information, which can replace the tedious work of manual labeling. Figure 3 illustrates some examples of synthesized defect samples.

3.3. Defect Repairing. Considering the real-time requirements of industrial production, we designed a simplified SegNet model. Compared with FCN [23] and U-Net [26], SegNet [29] uses the position information during maximum pooling. This does not require learning and, therefore, reduces the number of end-to-end training parameters. SegNet cleverly achieves upsampling by recording the position of the maximum value during pooling, and because there is no deconvolution process, it improves the training speed of the model.

In this study, we propose a concise SegNet with a reduced number of network layers. As shown in Figure 4,

the model uses fewer coding and decoding layers but can retain more detail for repairing the image while also significantly reducing the storage space occupied by the model. In the encoding process, convolution and maximum pooling are alternately used to complete the downsampling of the image. This process is followed only three times (the original SegNet involves five downsampling times). In the decoding process, maximum depooling and convolution are alternately used and are performed only three times. Furthermore, the LeakyReLU activation function was used directly for the output. Pooling indices (location information during pooling) are used to transfer the decoder, record the location information during pooling, and directly place the value back to the original location for unpooling.

Figure 5 shows the structure of the discriminant model, which uses a six-layer convolutional encoder structure. After

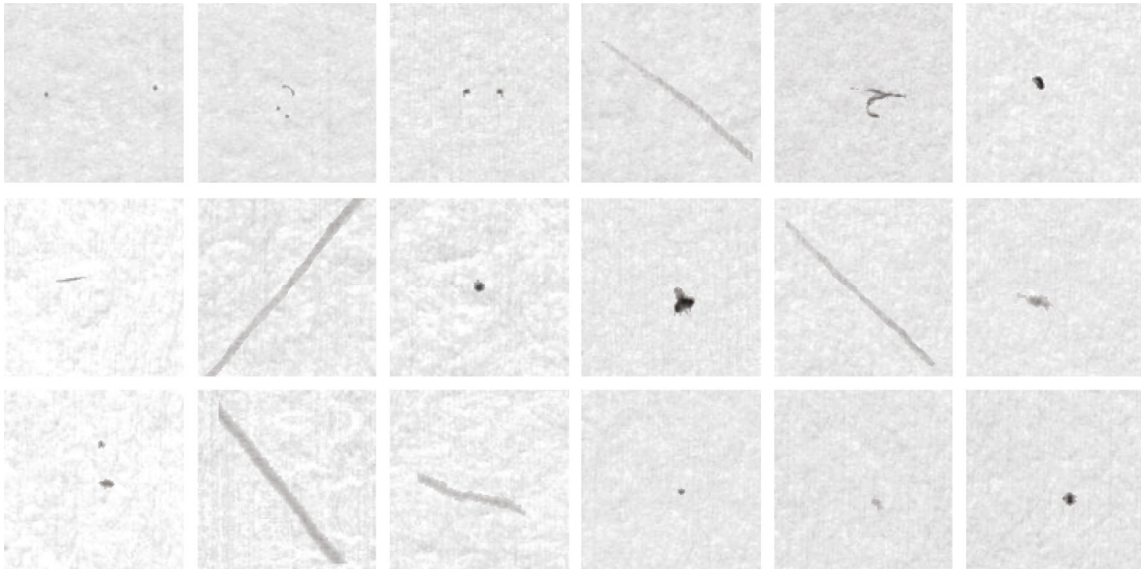


FIGURE 3: Examples of synthesized defect samples.

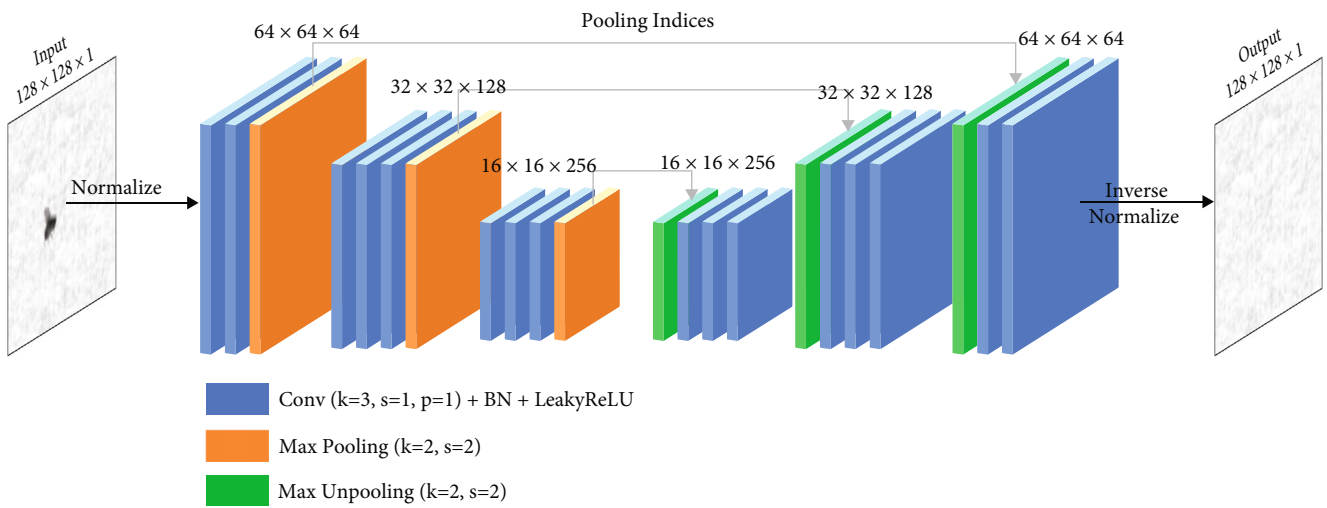


FIGURE 4: Generated network structure.

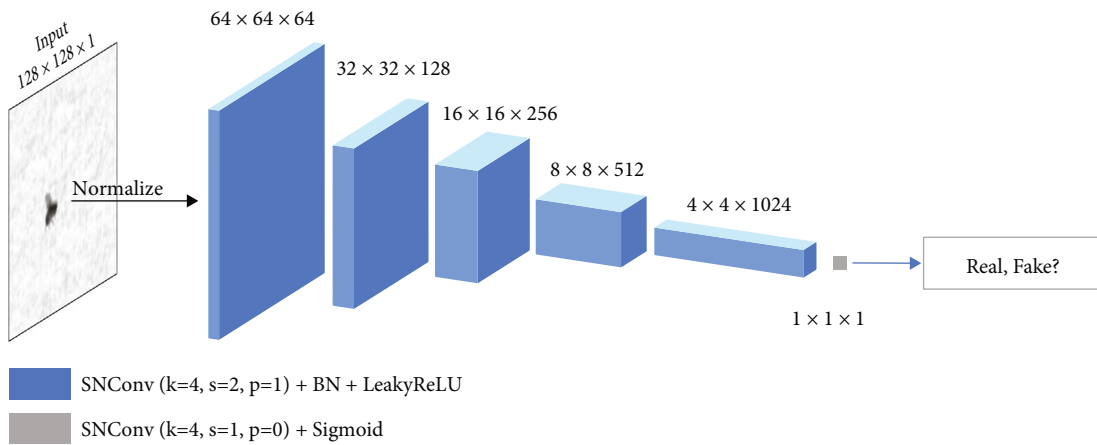


FIGURE 5: Discriminant network structure.

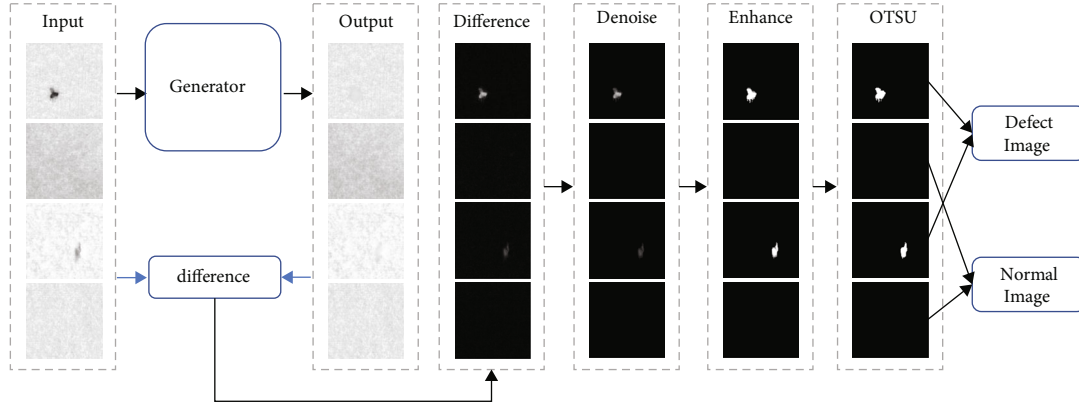


FIGURE 6: Process of segmenting defect images.

convolution, it is activated by the LeakyReLU function, and the last output layer uses the sigmoid function. Then, a score is obtained to determine whether the input image is truly normal based on the probability value.

3.4. Defect Segmentation. The end-to-end defect repair model is finally obtained through alternate training of the G and D networks. The test samples were inputted into the generated network to obtain a normal image.

There are two situations in this study: if there are any defects in the input sample, the model repairs the defects; otherwise, there is no significant difference between the output of the model and the input if the sample is normal. The original image and the repaired image need to be compared to obtain the difference image. The image difference can be described by formula (3).

$$\text{Dif}(x, y) = |I_{\text{ori}}(x, y) - I_{\text{rep}}(x, y)|, \quad (3)$$

where $I_{\text{ori}}(x, y)$ is the original image, $I_{\text{rep}}(x, y)$ the repaired image, and $|\cdot|$ the absolute value sign.

Since the defect area in the difference image may not be apparent, several enhancement operations are required, as shown in Figure 6. Conventional filtering methods are not used for denoising because they may blur the edges and details of the target. Rather, the threshold method is used to denoise the image directly, based on the background of the difference image. While filtering out the noise, the details of the segmented target can be preserved. The threshold method can be described by formula (4).

$$I_{\text{deno}}(x, y) = \begin{cases} \text{Dif}(x, y), & \text{if } \text{Dif}(x, y) \geq \text{th}, \\ 0, & \text{else.} \end{cases} \quad (4)$$

where th is a threshold for denoising.

Then, the brightness and contrast of the difference images are enhanced by a linear transformation, as in formula (5).

$$Y = \alpha X + \beta, \quad (5)$$

where X represents the pixel value of a certain point in the

original image and Y represents the pixel value of the corresponding position after transformation. The contrast of the image can be adjusted using α , and the brightness of the image can be changed using β .

Finally, the OTSU algorithm is used for binarization to obtain the required mask image, which is the final segmentation result.

4. Training Process and Model Optimization

4.1. Training Process. In the sample-synthesizing module, we obtain the image group consisting of the defect image, repaired image, and mask image. Only the first two were used in the training of the repair model. In the GAN training, an alternate iteration method is adopted for model training. First, the G network was trained, following which the D network was trained. The training of the D network also requires the output of the G network in the previous round of gradient backpropagation as input. Figure 7 shows the training process for the G and D networks.

For the G network, the defect samples are input into the generation model to generate a fake repair image, and then, the discriminant model is used to obtain a score. The expected repair image generated is sufficiently real; therefore, this score will form an error with the true label "1." Meanwhile, an error is formed between the false and true repair images generated. The aforementioned two errors are combined to form the loss function of the G network, and the parameters of the G network can be updated by the gradient backpropagation through the loss function.

For the training process of the D network, a score was obtained after the true repaired image was inputted into the discriminant model. The D network is expected to be able to accurately distinguish between true and false repaired images. Therefore, the discriminant score of the true repaired image and the true label "1" form an error. Similarly, the score of the false repaired image and the false label "0" form an error. The average of the two errors constitutes the loss function of the D network.

The role of the D network is to interfere with the generation model, that is, the score of the true repaired image tends to the true label "1" and the score of the false repaired image tends to the false label "0." This contradicts the

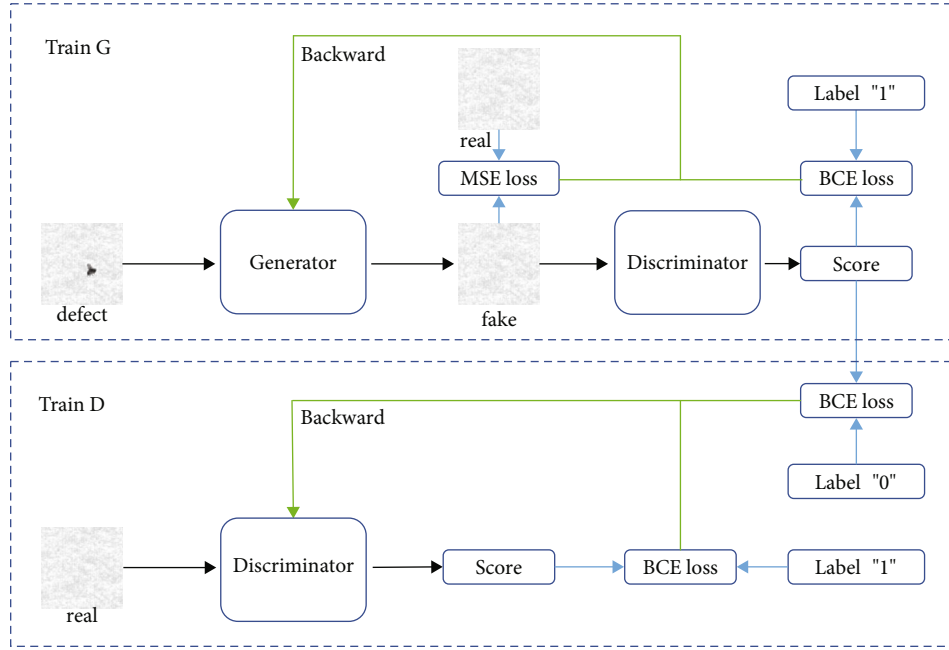


FIGURE 7: Training process of defect repair model.

expectation of the G network that the score of the fake repaired image tends to the true label “1,” which is the antagonism of the GAN model. In an ideal situation, when the scores obtained after the true and false repaired images entering the discriminant model are all close to 0.5, it means that the discriminant model is unable to distinguish between the true and false repaired images. This means that the sample generated by the generation model has become the data of the real sample distributed. At this time, the model reaches an ideal balance.

4.2. Objective Function. In the training of the adversarial segmentation network, there are four errors: the discriminant error and the generation error of the G network and two discriminant errors of the D network. Therefore, four loss functions were included in the error analysis. For the G network, there was an error between the false repaired image and the true repaired image. The mean square error (MSE) was used for evaluation. In addition, there is an error between the score of the fake repaired image and the true label “1.” This is a binary classification problem. Binary cross entropy (BCE) was used to calculate the loss. Similarly, in the D network, both errors were binary classification problems and BCE was used to calculate the loss.

First, we observe the composition of the MSE loss function, as shown in formula (6). An additional sample number average compared to the Euclidean distance formula can be described as the expected value of the square of the difference between the true value and the estimated value. The MSE loss of the G network can be simply described by formula (7).

$$\text{MSE}(y_i, \hat{y}_i) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (6)$$

where y_i and \hat{y}_i are the true and estimated values, respectively.

$$\text{MSE loss} = E[y - G(x)]^2, \quad (7)$$

where x and y are the defect and true repaired samples, respectively.

The calculation of the BCE loss function is described in formula (8), where \hat{y}_i is the evaluation value of the sample and y_i is the label of the binary classification, which is 0 or 1.

$$\text{BCE}(\hat{y}_i, y_i) = \frac{1}{n} \sum_{i=1}^n [-y_i \log \hat{y}_i - (1 - y_i) \log (1 - \hat{y}_i)], \quad (8)$$

$$\text{BCE}(\hat{y}_i, 1) = \frac{1}{n} \sum_{i=1}^n [-\log \hat{y}_i], \quad (\text{if label } y_i = 1), \quad (9)$$

$$\text{BCE}(\hat{y}_i, 0) = \frac{1}{n} \sum_{i=1}^n [-\log (1 - \hat{y}_i)], \quad (\text{if label } y_i = 0).$$

It can be concluded that the objective function of the D network can be expressed as

$$\min V(D) = \frac{1}{2} E[-\log (D(y))] + \frac{1}{2} E[-\log (1 - D(G(x)))]. \quad (10)$$

The objective function of the G network can be described as

$$\min V(G) = (\lambda - 1) E[y - G(x)]^2 + \lambda E[-\log (D(G(x)))], \quad (11)$$

Then, to unify the formula, the objective function of the D network can be changed to

$$\max V(D) = \frac{1}{2}E[-\log(1 - D(y))] + \frac{1}{2}E[-\log(D(G(x)))]. \quad (12)$$

That is, the final objective function of the trained model can be described as

$$\min_G \max_D V(G, D) = (\lambda - 1)E[y - G(x)]^2 + \lambda(E[-\log(1 - D(y))] + E[-\log(D(G(x)))]). \quad (13)$$

4.3. Model Optimization. Aiming at the fabric sample characteristics, especially weak and small targets, some optimizations were performed on the model.

- (1) As mentioned earlier, the model in this study adopts the largest depooling layer in SegNet because, compared to deconvolution, the amounts of calculation and space occupation are less. Meanwhile, fewer coding and decoding layers are used but can retain more details to repair the image and significantly reduce the storage space occupied by the model. The downsampling process uses only eight layers of convolution and three layers of maximum pooling. Meanwhile, the multiclass SoftMax output layer was removed, making the output come directly after convolution. This simplifies the model and can meet the needs of an industrially embedded system
- (2) Instance normalization (IN) is used instead of regular batch normalization (BN) in the defect repair network because IN is suitable for repairing defect images to normal images in the generator model. Since the result of image generation mainly depends on a certain image instance, using IN not only accelerates the model convergence but also maintains the independence between each image instance
- (3) This model uses LeakyReLU as the activation function. For a regular ReLU activation function, when the input value is negative, the output value would be zero. Since the training goal is to obtain a repaired image, which has a similar value range as the input defect image, it needs to be activated as a negative value when the input is negative. Therefore, choosing LeakyReLU as the activation function not only solves the problem that ReLU can easily lead to necrosis but also ensures that the information is not completely lost when the input is negative. Therefore, the defect repair model can generate a better repair result

5. Performance Analysis

5.1. Experimental Setup. The experimental environment used was PyTorch1.3.1, Windows 10 system, CUDA 10.1,

TABLE 1: Experimental setup.

Parameter	Setting
Batch size	32
Training epoch	100
Initial learning rate	0.001
Momentum	0.5/0.999
Learning rate	0.0001/0.00001

GPU: GTX 1050ti, and cuDNN 7.0. The sample resolution was 128×128 pixels.

The main parameter settings of the experiment are shown in Table 1.

All data in each epoch went through the network. In the training process, this experiment used the Adam optimizer, which combines the advantages of the RMSProp and Ada-Grad optimization algorithms. In this experiment, the initial learning rate set for the Adam optimizer was 0.001 and the momentum values of the first-order moment and second-order moment estimation were 0.5 and 0.999, respectively. Meanwhile, a multistep learning rate decay strategy (Multi-StepLR) was set. In the experiment, the learning rate was 0.0001 when the epoch was 30 and the learning rate was 0.00001 when the epoch was 60. The advantage of this setting is that the loss of the model can be rapidly reduced in the early stage and can gradually reach the optimum in the later stage.

Image processing after defect repair was implemented using the OpenCV method; the denoising threshold was set to 19, linear transformation process used the convertScaleAbs method, and α and β parameters were set to 5 and 0, respectively.

5.2. Datasets. The experiments in this study used the following three datasets:

5.2.1. Enterprise Dataset. The fabric defect samples in the experiment originated from the image acquisition equipment on the enterprise assembly line. In the production process, high-speed cameras are used to monitor product quality.

There were 4360 original samples, and the original image resolution was 371×257 pixels. After removing duplicate and invalid samples, there were only a total of 90 samples and these were labeled for defects. Then, we used image rotation, flip, transpose, and other operations to enlarge the image set and obtain seven new forms of defect images. In the process of transforming the defect image, the label corresponding to the image is also expanded so that there is no need to label the new defect image one by one.

We then adopted the method described in Section 3, to quickly obtain a large number of defect samples that were close to the original defect sample distribution, where the “sliding cutting” method was used with a sliding step of 20 and cutting resolution of 128×128 pixels.

5.2.2. AITEX Dataset. The AITEX dataset [57] is composed of 245 images of 4096×256 pixels with seven different fabric structures. There are 140 nondefect images in the database

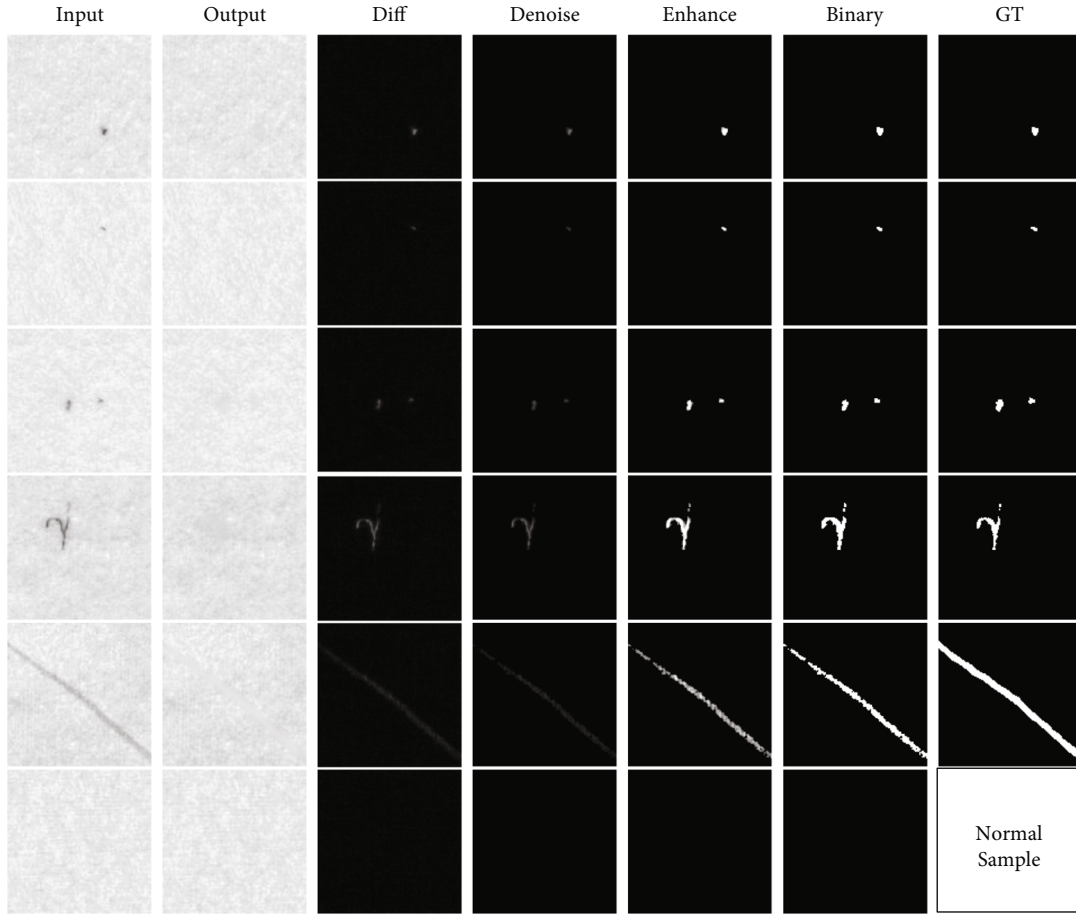


FIGURE 8: Segmentation results of enterprise data (the first five rows are defect samples, and the last row lists normal samples).

and 105 images of 12 different types of fabric defects that are common in the textile industry.

5.2.3. Expanded Dataset. Because the defect samples are actually small, the existing defects and types of defects are very limited. Therefore, we created artificial defect samples. Such defects did not appear in the training set and were, therefore, used to test whether our defect segmentation model was effective.

5.3. Evaluation Metrics. Our model is evaluated using several metrics, such as Pixel Acc and intersection over union (IoU). Pixel Acc represents the ratio of the number of correctly classified pixels to the total number of pixels in the segmentation image, including correctly classified background points. The IoU measures the similarity between the segmentation result and the ground truth, as shown in formula (14). Each pixel in the segmentation result is divided into four types, that is, true positive (TP) (the number of defect pixels that are correctly divided into the defect area by the model), false positive (FP) (the number of background pixels that are incorrectly divided into the defect area by the model), false negative (FN) (the number of defect pixels that are incorrectly divided into the background area by the model), and true negative (TN) (the number of background

pixels that are correctly divided into the background area by the model).

$$\text{IoU} = \frac{TP}{TP + FN + FP}. \quad (14)$$

We used the mean absolute error (MAE) to evaluate the average pixel error after image repair, as in formula (15).

$$\text{MAE} = \frac{\sum_{i=1}^n |A_i - C_i|}{n}, \quad (15)$$

where A_i is the original value of the i th pixel, C_i is the repaired value of the i th pixel, and n is the total number of pixels in an image.

In addition, we used the structural Ssmilarity Iidex (SSIM) [58] to analyze the quality of image restoration.

5.4. Test Results

5.4.1. Enterprise Fabric Samples. Through the sample-synthesizing module, we obtained 19606 pairs of artificial defect samples. A total of 10240 pairs of samples were used as the training set, and 320 pairs were used as the validation set to observe the effect that training had on the model. The

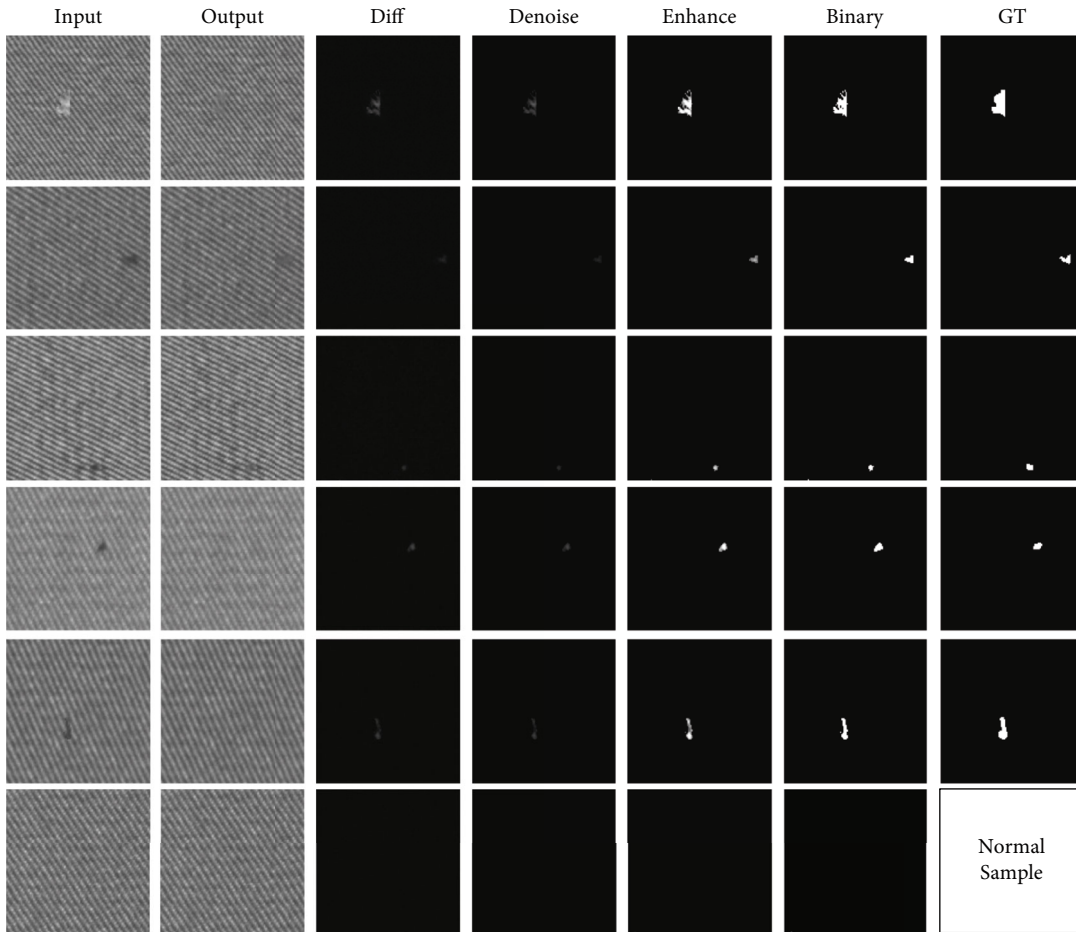


FIGURE 9: Segmentation results of AITEX samples (the first five rows are defect samples, and the last row lists normal samples).

TABLE 2: A comparison of segmentation results of different models.

	Pixel Acc	IoU
FCN [23]	0.9944	0.5811
U-Net [26]	0.9959	0.6400
SegNet [29]	0.9963	0.6811
FCNGAN	0.9952	0.5876
U-NetGAN	0.9951	0.6367
SegNetGAN	0.9968	0.7034
Ours	0.9968	0.7784

test set used the original 720 defect samples and 6480 normal samples.

In the fabric defect samples, the optimal model-generated error (MSE error) in the validation set was only 0.00021. The Pixel Acc was 99.68%, and IoU accuracy was 77.84%. Figure 8 shows the segmentation results for the samples.

5.4.2. AITEX Dataset. First, the large-sized samples were cropped to obtain 128×128 pixel samples. The sample preprocessing method and training hyperparameter settings were the same as those of the previous fabric defect sample set.

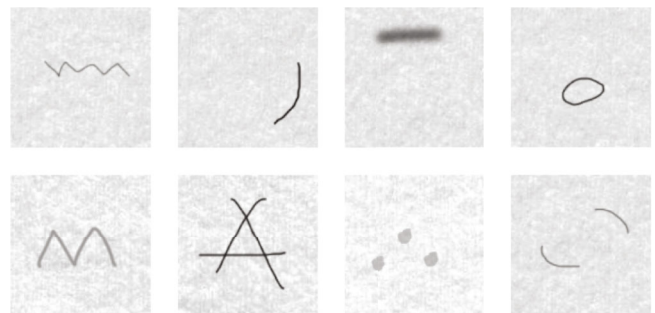


FIGURE 10: Defect samples by artificial construction.

The optimal model-generated error (MSE error) on the validation set was 0.00056, the Pixel Acc of defect segmentation was 99.94%, and the IoU score was 77.85%. Figure 9 illustrates the segmentation results for the AITEX samples. The model in this study also has a good effect on this type of model with a more complex background in terms of segmentation accuracy. Table 2 lists the experimental results for several samples in this model.

5.4.3. Extended Dataset. There would be some undetected and excluded defects in the training because defects of an

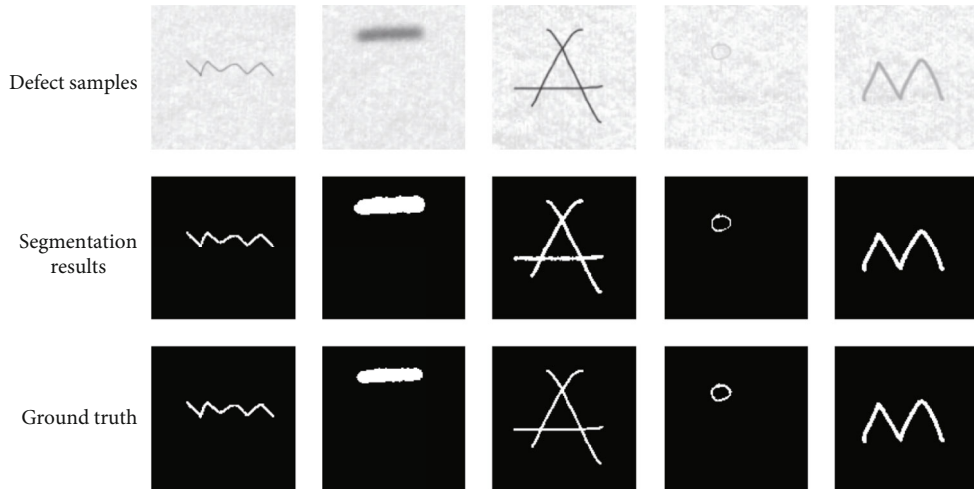


FIGURE 11: Segmentation results of artificially defected samples.

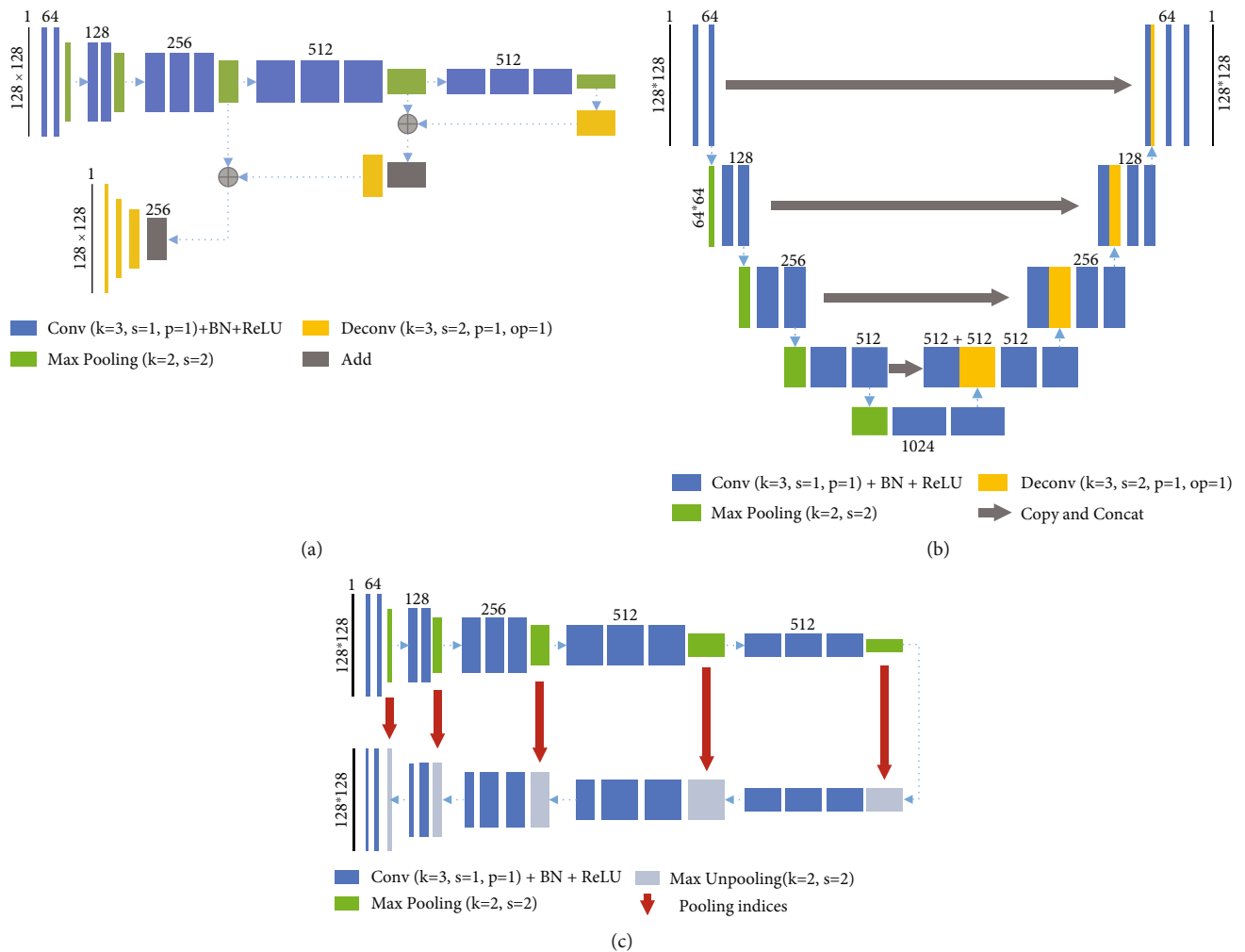


FIGURE 12: Three models in Table 2. (a) FCN, (b) U-Net, and (c) SegNet.

unknown type may be present in the production process. Several new types of defect samples were artificially constructed to test the robustness of our proposed model, as shown in Figure 10. These types of defects did not appear

in the previous training samples and test samples. Examples of such defects are large-area defects and long-line defects.

After testing the extended sample set, the results showed that the segmentation effect of our previous model was

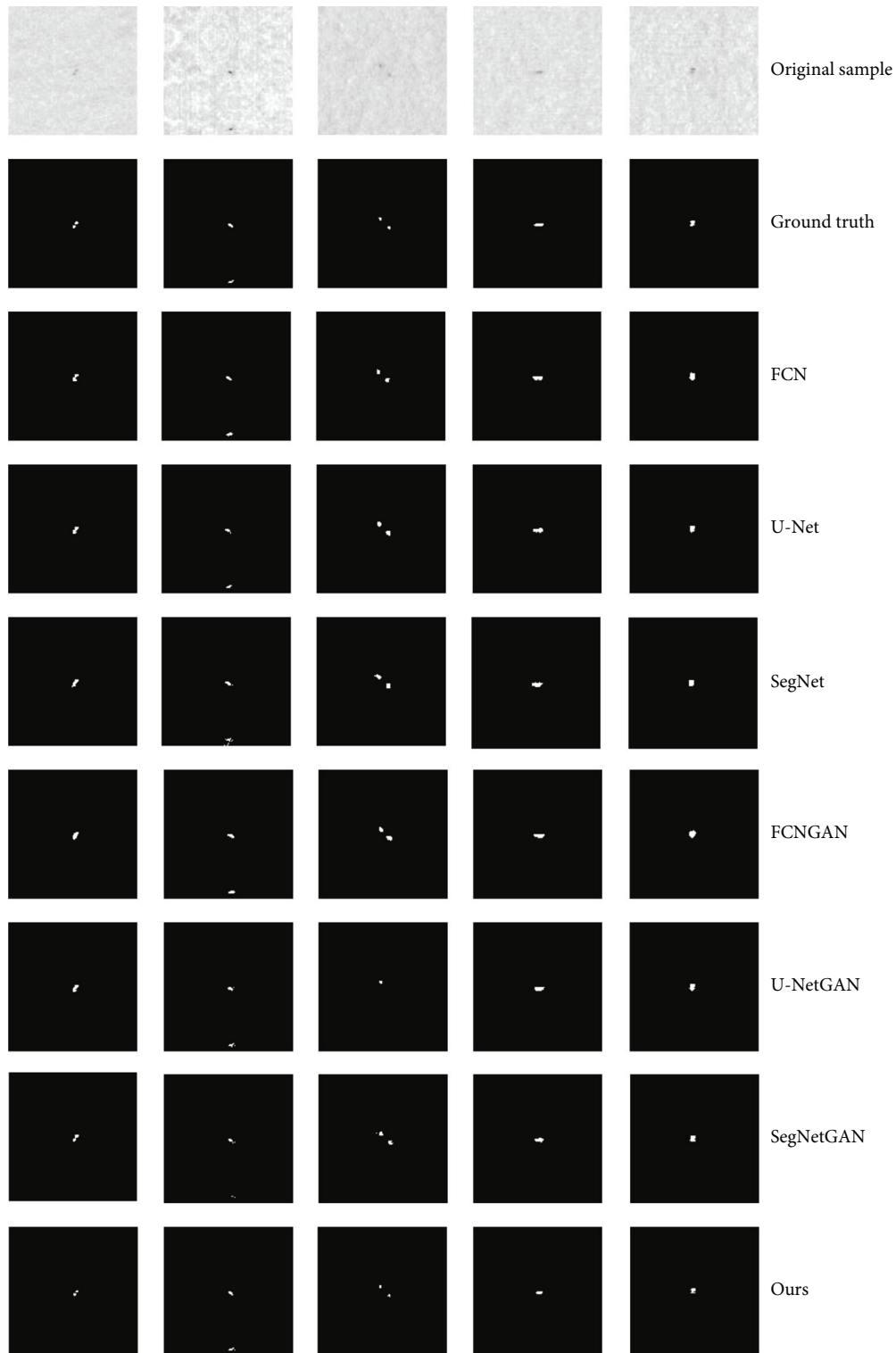


FIGURE 13: Segmentation results of weak and small-defect samples.

excellent. Figure 11 illustrates examples of the artificial defect segmentation results. In the extended set of 64 samples, the Pixel Acc of segmentation reached 99.3% and the IoU score reached 73.6%. The results achieved the segmentation accuracy of the existing defect samples.

5.5. Analysis

5.5.1. Comparative Experiments. To compare the performance of the different models, we implemented six other models, as shown in Table 2.

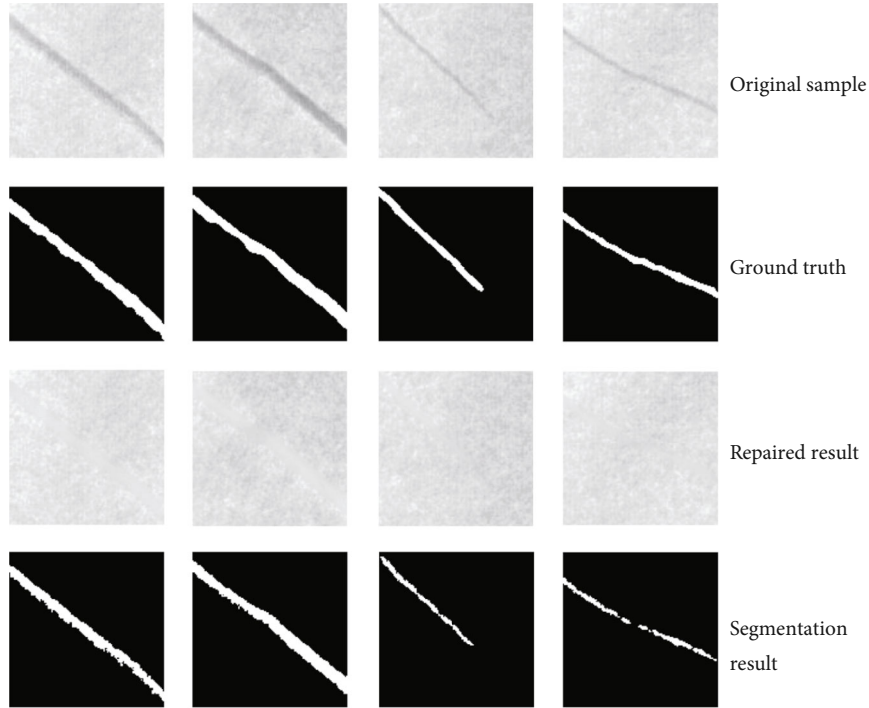


FIGURE 14: Segmentation results of uneven background samples.

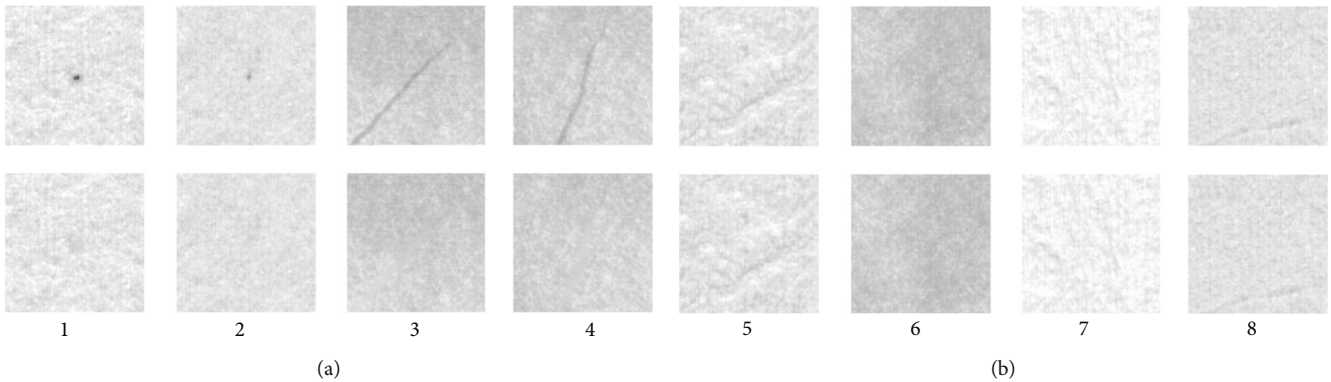


FIGURE 15: Repair results. (a) Repair results of the defective sample (the first row lists the original samples, and the second row lists the repaired results). (b) Repair results of the normal sample (the first row lists the original samples, and the second row lists the repaired results).

TABLE 3: MAE of repaired results in Figure 15.

Sample	1	2	3	4	5	6	7	8
MAE	1.22	0.87	2.18	1.83	0.98	1.65	1.28	0.85

TABLE 4: SSIM results in Figure 15.

Sample	1	2	3	4	5	6	7	8
SSIM	0.981	0.986	0.941	0.953	0.989	0.976	0.981	0.989

The FCN, U-Net, and SegNet models are described in [23, 26, 29], respectively. The specific implementation details of the three models are presented in Figure 12.

FCNGAN, U-NetGAN, and SegNetGAN in Table 2 indicate a model obtained by training based on the GAN mechanism, where FCN, U-Net, and SegNet, respectively,

TABLE 5: A comparison of model size and FPS for various architectures.

	Model size (MB)	FPS
FCN [23]	89.2	132
U-Net [26]	131.7	69
SegNet [29]	112.4	109
FCNGAN	89.2	131
U-NetGAN	131.7	69
SegNetGAN	112.4	106
Ours	14.4	107

are used as the G network and the D network is composed of a six-layer convolutional network. The D-network model is shown in Figure 5. The training parameters of the models listed in Table 2 are consistent with those listed in Table 1.

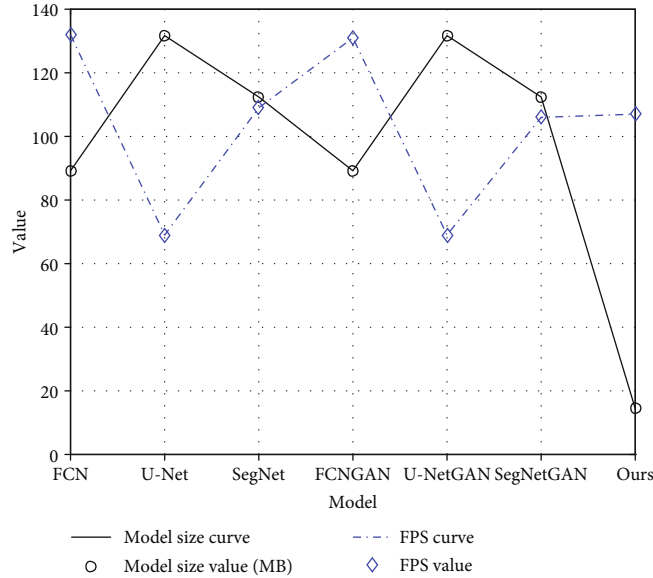


FIGURE 16: Comparison of the model size and FPS for various architectures.

TABLE 6: Segmentation results of different resolution.

Resolution	Pixel Acc	IoU	FPS
256×256	0.9985	0.7754	37
128×128	0.9968	0.7784	107

The Pixel Acc of the above models exceeded 0.99. The IoU indicator of the abovementioned model widened the gap, and the gap between high and low reached approximately 20%.

Among the first three segmentation models, SegNet has more advantages and the three segmentation evaluation indicators are better than the other two models. After the GAN training mechanism was introduced into the three segmentation models, the segmentation performance of FCNGAN and SegNetGAN was improved but U-NetGAN did not. Through comparison, it was found that the GAN training mechanism did not significantly improve the performance of the three semantic segmentation models. The model proposed herein achieved the best experimental results, and the segmentation performance evaluation index was better than the other six models. In this experiment, many of the sample defect areas were weak and small targets; therefore, the fluctuation of the segmentation results had a greater impact on the IoU but the IoU reached 0.7784, which is 7.5% higher than the best result of 0.7034 in the other six models.

5.5.2. Segmentation Effects of Weak and Small-Defect Samples.

We compared the segmentation results of each model for weak and small-defect samples. As shown in Figure 13, the first row contains five weak and small-defect samples and the second row is the ground truth. Upon comparison, it was found that the first sample on the left contained two very small defects situated very close. The model segmentation

result is closest to the ground truth, which separates two small defects.

In another example, there were two small defects in the third sample. The segmentation result of the U-NetGAN model misses the defect, and the defect segmentation results of the other models are enlarged. The segmentation results of the model proposed in this study are the most accurate.

5.5.3. Samples with Uneven Background. To test the ability of the model proposed herein to repair defect samples, we selected some defect samples with uneven backgrounds for testing. The test results show that the proposed model can effectively segment the defects. Since our model uses an image difference algorithm, the defect area that is very similar to the background may not be continuous in the segmentation results, as illustrated in Figure 14.

5.5.4. Analysis of the Sample Repair Effect. This study provides representative samples for analyzing the repair results of our model. In Figure 15(a), sample 1 has an apparent flaw, sample 2 has a weak and small defect, and samples 3 and 4 have long stripe defects. These four samples in Figure 15(b) illustrate normal samples with different backgrounds and textures.

Figure 15(a) illustrates the result of repairing the four defect samples. From the results of the repair, the flaws assumingly disappeared. We used the MAE to evaluate the average pixel error after image repair, as shown in Table 3. From the MAE results, the pixel error of the repaired image was less than three. Since samples 3 and 4 had long strips of flaws with larger areas, the MAE was also larger.

Figure 15(b) illustrates the results of normal sample image restoration. The results generated by our model are nearly identical to those of the original images. The MAE results showed that the average pixel error was less than two. The results showed that the repair effect of our model was excellent.

Moreover, we used SSIM to analyze the quality of image restoration. Table 4 presents the SSIM results, which show that the similarity between the original and repaired samples was generally high. In the sample in Figure 15(a) with flaws, samples 1 and 2 have a minimal effect on the similarity because the flaws are small, whereas, in samples 3 and 4, the similarity decreases owing to the larger area of the flaws. For the normal sample in Figure 15(b), the similarity was high. This indicates that the repair quality of our model is good.

5.5.5. Model Size Comparison. To verify whether the model can meet real-time requirements, we tested the segmentation speed of seven different models. The research index is the number of cotton samples (frames per second (FPS)) that the model can process in one second. The experimental results are presented in Table 5 and Figure 16. The model size represents the size of the saved model file. The model proposed in this article occupies a small space, only 14.4MB, which is easy to be embedded in industrial equipment.

The first three models follow the probability that the smaller the model, the higher the FPS value. This is because the model size and computing speed do not necessarily show an anticorrelation. The size of the model directly represents the number of parameters of the model, but the speed of the model calculation is not only related to the number of parameters but also affected by the structure of the model. The processing speeds of the seven models can meet real-time requirements. Since our model is calculated using a GPU, the process of repairing the network to obtain the repaired image is very fast: the processing of 7200 samples takes only approximately 56 s (equal to 128 FPS). As for the image processing operations after repairing the network, the calculation time is only slightly increased based on the OpenCV calculation on the CPU. This results in a decrease in the overall FPS but still achieves good real-time performance.

Considering that the resolution of the test samples has an impact on the FPS, we used a sample set with a resolution of 256×256 pixels to test the model again; this included 720 defect samples and 6480 normal samples. These test samples did not come from an enlarged 128×128 -pixel image but were cut directly from the original cotton cloth sample. The test results are presented in Table 6. The test results show that the FPS decreased owing to the increase in sample resolution; however, the real-time requirements can still be reached.

6. Conclusion

In this study, a lightweight system composed of three modules was designed to solve the segmentation problem of fabric defects, particularly for weak and small-defect targets. We used a GAN model based on the repair mechanism, which is lightweight and has good defect segmentation ability. The results of testing corporate samples and samples from a public database show that the model proposed in this study has good segmentation effects and can achieve real-

time performance, thus demonstrating its application value in IIoT and industrial production lines.

In the future, we will focus on few-shot and unsupervised learning. In addition, further improvements in real-time performance are worth studying.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the project of the National Natural Science Foundation of China (Grant 61772562 and Grant 61976226) and the Hubei Provincial Natural Science Foundation of China for distinguished young scholars (Grant 2017CFA043). The authors thank Mr. Zheng He for the original data and Dr. Jingang Yu for the constructive discussions and suggestions.

References

- [1] S. Wan, S. Ding, and C. Chen, "Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles," *Pattern Recognition*, vol. 121, article 108146, 2022.
- [2] J. Yang, C. Wang, B. Jiang, H. Song, and Q. Meng, "Visual perception enabled industry intelligence: state of the art, challenges and prospects," *IEEE Trans. Ind. Inform.*, vol. 17, no. 3, pp. 2204–2219, 2021.
- [3] A. Kumar, "Computer-vision-based fabric defect detection: a survey," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 1, pp. 348–363, 2008.
- [4] K. Hanbay, M. F. Talu, and Ö. F. Özgüven, "Fabric defect detection systems and methods—a systematic literature review," *Optik*, vol. 127, no. 24, pp. 11960–11973, 2016.
- [5] A.-A. Tulbure, A.-A. Tulbure, and E.-H. Dulf, "A review on modern defect detection models using DCNNs – deep convolutional neural networks," *Journal of Advanced Research*, vol. 35, pp. 33–48, 2022.
- [6] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, "Automated fabric defect detection—A review," *Image and Vision Computing*, vol. 29, no. 7, pp. 442–458, 2011.
- [7] F. S. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 803–808, 1991.
- [8] C.-H. Chan and G. K. H. Pang, "Fabric defect detection by Fourier analysis," *IEEE Transactions on Industry Applications*, vol. 36, no. 5, pp. 1267–1276, 2000.
- [9] L. Tong, W. K. Wong, and C. K. Kwong, "Differential evolution-based optimal Gabor filter model for fabric inspection," *Neurocomputing*, vol. 173, pp. 1386–1401, 2016.
- [10] Z. Zhan, J. Zhou, and B. Xu, "Fabric defect classification using prototypical network of few-shot learning algorithm," *Computers in Industry*, vol. 138, article 103628, pp. 1–11, 2022.

- [11] X. Zheng, S. Zheng, Y. Kong, and J. Chen, "Recent advances in surface defect inspection of industrial products using deep learning techniques," *The International Journal of Advanced Manufacturing Technology*, vol. 113, pp. 35–58, 2021.
- [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 3, pp. 2672–2680, 2014.
- [13] Z. Zhao, B. Li, R. Dong, and P. Zhao, "A surface defect detection method based on positive samples," in *Lecture Notes in Computer Science Pac. Rim International Conference on Artificial Intelligence*, pp. 473–481, Springer, Cham, 2018.
- [14] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1840–1852, 2020.
- [15] D.-M. Tsai and C.-H. Chiang, "Automatic band selection for wavelet reconstruction in the application of defect detection," *Image and Vision Computing*, vol. 21, no. 5, pp. 413–431, 2003.
- [16] H. Y. T. Ngan, G. K. H. Pang, S. P. Yung, and M. K. Ng, "Wavelet based methods on patterned fabric defect detection," *Pattern Recognition*, vol. 38, no. 4, pp. 559–576, 2005.
- [17] H. Zhang, J. Ma, J. Jing, and P. Li, "Fabric defect detection using l0 gradient minimization and fuzzy C-means," *Applied Sciences*, vol. 9, no. 17, pp. 3506–3516, 2019.
- [18] Q. Wang, J. Jing, L. Zhang, and X. Wang, "Denim defect detection based on optimal Gabor filter," *Laser & Optoelectronics Progress*, vol. 55, no. 7, article 071501, 2018.
- [19] D. Yapi, M. S. Allili, and N. Baaziz, "Automatic fabric defect detection using learning-based local textural distributions in the Contourlet domain," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1014–1026, 2018.
- [20] X. Kang and E. Zhang, "A universal defect detection approach for various types of fabrics based on the elo-rating algorithm of the integral image," *Textile Research Journal*, vol. 89, no. 21–22, pp. 4766–4793, 2019.
- [21] M. T. Habib, S. B. Shuvo, M. S. Uddin, and F. Ahmed, "Automated textile defect classification by Bayesian classifier based on statistical features," in *2016 International Workshop on Computational Intelligence (IWCI)*, pp. 101–105, Dhaka, Bangladesh, 2016.
- [22] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimize XGBoost training parameters," *IEEE Access*, vol. 7, pp. 118310–118321, 2019.
- [23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, USA, 2015.
- [24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.
- [26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.
- [27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, 2016.
- [28] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1925–1934, Honolulu, HI, USA, 2017.
- [29] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [30] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.
- [31] Y. Li, W. Zhao, and J. Pan, "Deformable patterned fabric defect detection with Fisher criterion-based deep learning," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1256–1264, 2017.
- [32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, Munich, Germany, 2018.
- [33] L. Wang, H. Zhen, X. Fang, S. Wan, W. Ding, and Y. Guo, "A unified two-parallel-branch deep neural network for joint gland contour and segmentation learning," *Future Generation Computer Systems*, vol. 100, pp. 316–324, 2019.
- [34] Y. Zhao, K. Hao, H. He, X. Tang, and B. Wei, "A visual long-short-term memory based integrated CNN model for fabric defect image classification," *Neurocomputing*, vol. 380, pp. 259–270, 2020.
- [35] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10778–10787, Seattle, WA, USA, 2020.
- [36] H. Wang, D. Zhang, S. Ding, Z. Gao, J. Feng, and S. Wan, "Rib segmentation algorithm for X-ray image based on unpaired sample augmentation and multi-scale network," *Neural Computing and Applications*, pp. 1–15, 2021.
- [37] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 929–940, 2018.
- [38] D. Tabernik, S. Šela, J. Skvařc, and D. Sko, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020.
- [39] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, article 105124, 2020.
- [40] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, Salt Lake City, UT, USA, 2018.
- [41] P. Wang and X. Bai, "Thermal infrared pedestrian segmentation based on conditional GAN," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6007–6021, 2019.

- [42] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in *Proceedings of the European conference on computer vision (ECCV)*, vol. 11213, pp. 186–201, Munich, Germany, 2018.
- [43] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, Long Beach, CA, USA, 2019.
- [44] B. J. B. Rani, "Survey on applying GAN for anomaly detection," in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, Coimbatore, India, 2020.
- [45] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," in *International Conference on Information Processing in Medical Imaging. Lecture Notes in Computer Science*, vol. 10265, pp. 146–157, Springer, Cham, 2017.
- [46] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-Supervised Anomaly Detection Via Adversarial Training," in *Asian Conference on Computer Vision*, pp. 622–637, Springer, Cham, 2019.
- [47] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: one-class novelty detection using GANs with constrained latent representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019*, pp. 2893–2901, Long Beach, CA, USA, 2019.
- [48] P. C. Ngo, A. A. Winarto, C. K. Kou, S. Park, F. Akram, and H. K. Lee, "Fence GAN: Towards better anomaly detection," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 141–148, Portland, OR, USA, 2019.
- [49] J. Wang, W. Huang, S. Wang, P. Dai, and Q. Li, "LRGAN: visual anomaly detection using GAN with locality-preferred recoding," *Journal of Visual Communication and Image Representation*, vol. 79, pp. 103201–103208, 2021.
- [50] S. Nema, A. Dudhane, S. Murala, and S. Naidu, "RescueNet: an unpaired GAN for brain tumor segmentation," *Biomedical Signal Processing and Control*, vol. 55, pp. 101641–101648, 2020.
- [51] B. Murugesan, K. Sarveswaran, S. M. Shankaranarayana, K. Ram, and M. Sivaprakasam, "A context based deep learning approach for unbalanced medical image segmentation," in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 1949–1953, Iowa City, IA, USA, 2020.
- [52] J. Liu, C. Wang, H. Su, B. Du, and D. Tao, "Multistage GAN for fabric defect detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 3388–3400, 2019.
- [53] Y. Huang, J. Jing, and Z. Wang, "Fabric defect segmentation method based on deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.
- [54] C. H. Yeh, C. H. Lin, L. W. Kang et al., "Lightweight deep neural network for joint learning of underwater object detection and color conversion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, pp. 1–15, 2021.
- [55] Y. Liu, X. Y. Zhang, J. W. Bian, L. Zhang, and M. M. Cheng, "SAMNet: stereoscopically attentive multi-scale network for lightweight salient object detection," *IEEE Transactions on Image Processing*, vol. 30, pp. 3804–3814, 2021.
- [56] Z. Cao, X. Xu, B. Hu, and M. Zhou, "Rapid detection of blind roads and crosswalks by using a lightweight semantic segmentation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6188–6197, 2021.
- [57] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, and J. Moreno, "A public fabric database for defect detection methods and results," *Autex Research Journal*, vol. 19, no. 4, pp. 363–374, 2019.
- [58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.