Article

Automatic Unsupervised Fabric Defect Detection Based on Self-Feature Comparison

Zhengrui Peng [1], Xinyi Gong [2], Bengang Wei [1], Xiangyi Xu [1] and Shixiong Meng [3,*]

[1] State Grid Shanghai Electric Power Research Institute, Shanghai 200437, China; skanson@163.com (Z.P.); wbgsj@126.com (B.W.); xuxiangyi_1986@126.com (X.X.)

[2] Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; gongxinyi2014@ia.ac.cn

[3] School of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing 100083, China

[*] Correspondence: henryxiong2021@163.com

Abstract: Due to the huge demand for textile production in China, fabric defect detection is particularly attractive. At present, an increasing number of supervised deep-learning methods are being applied in surface defect detection. However, the annotation of datasets in industrial settings often depends on professional inspectors. Moreover, the methods based on supervised learning require a lot of annotation, which consumes a great deal of time and costs. In this paper, an approach based on self-feature comparison (SFC) was employed that accurately located and segmented fabric texture images to find anomalies with unsupervised learning. The SFC architecture contained the self-feature reconstruction module and the self-feature distillation. Accurate fiber anomaly location and segmentation were generated based on these two modules. Compared with the traditional methods that operate in image space, the comparison of feature space can better locate the anomalies of fiber texture surfaces. Evaluations were performed on the three publicly available databases. The results indicated that our method performed well compared with other methods, and had excellent defect detection ability in the collected textile images. In addition, the visual results showed that our results can be used as a pixel-level candidate label.

Keywords: fabric defect; unsupervised learning; computer vision; deep learning

1. Introduction

The textile industry occupies a significant proportion of China's industry, and its products are widely used in homes, clothing, construction, and even aerospace. In the textile industry, the surface quality of products is an important factor that affects their price and grade evaluation [1]. Therefore, in the manufacturing process, the industry arranges an inspection process to ensure that flawless products are delivered to merchants or consumers. The traditional detection methods use a human to detect the surface defects, which is not only slow, but also cannot ensure the consistency of the detection effect. With the popularization of automatic product lines, automatic fabric defect detection equipment based on machine vision is increasingly being applied in fabric defect detection.

The core of surface defect detection based on industrial vision is to extract the features related to defects from fabric images. However, the visual inspection of surface defects is still difficult due to the diverse sizes, varying brightness, low contrast, complex features, and inadequate defect samples for different products. In the past few decades, researchers have proposed many surface defect inspection methods to overcome these difficulties. These traditional methods rely heavily on handcrafted features. In other

words, with the increase of defect types, these traditional methods that rely on prior experience will not satisfy the automation needs of the industry.

At present, the data-driven deep-learning method is widely used in many defect detection fields. This method can be divided into three categories: classification, detection, and segmentation [2]. Moreover, researchers have attempted to apply methods based on convolution neural networks to fabric defect detection in industrial settings. Jing et al. [3] proposed an extremely efficient convolutional neural network, Mobile-Unet [4], to achieve end-to-end fabric defect segmentation. Mobile-Unet introduced depthwise separable convolution, which dramatically reduced the complexity cost and model size of the network. Zhu et al. [5] designed the modified densenet [6] for automatic fabric defect classification with edge computing. Wu et al. [7] proposed a wide-and-light network structure based on Faster R-CNN for detecting common fabric defects. The feature extraction capability for fabric defects was enhanced by a multiscale dilated convolution kernel. In [8], a cascaded mixed feature pyramid network was proposed for guiding the localization of fabric defects. Although the above methods achieved good detection results, training these models was required to be supervised in the form of annotations. These annotations can be at the image level, bounding box level, or even pixel level, which will lead to some disadvantages. First, annotations in supervised methods often require an experienced inspector, and pixel-level annotations require a lot of time and costs. Secondly, these methods can only map existing defect types to known defect types, and cannot handle unknown types of anomalies. Finally, the difficulty of collecting flawless images is much lower than that of defective images in industrial sites. Hence, some fabric defect detection methods based on unsupervised learning have also attracted the interest of researchers, such as Markov random field [9], low-rank decomposition [10], and sparse dictionary [11]. However, these shallow-feature networks limit the detection ability of surface defects, resulting in the inability to adapt to complex and real industrial environments.

In this paper, we propose an unsupervised fabric defect detection method based on feature-compared training on defect-free samples. To avoid using a large number of training samples, feature extraction based on the pretrained model was applied. This approach could directly capture the normal variability of training data. Then, the obtained features went through the SFC architecture, which contained the self-feature reconstruction module (SRM) and the self-feature distillation (SFD). The surface defects were obtained by using combined judgment criteria based on feature reconstruction errors and feature distillation errors. Compared with the traditional methods operating in the image space, the comparison of feature space can better locate the anomalies of fiber texture surfaces. In this work, a combined feature reconstruction and feature distillation were utilized, and its normal features were learned simultaneously through a direct network and an indirect network.

The rest of this paper is organized as follows. In Section 2, the related works regarding unsupervised learning for anomaly detection are introduced and summarized. In Section 3, the proposed network is described and discussed in detail. In Section 4, experiments are designed to demonstrate the performance of our method, which includes the fabric datasets, experimental setups, results with comparative study, and extended application. Finally, we conclude the paper in Section 5.

2. Related Works

In this section, the related works regarding popular unsupervised learning for anomaly detection are introduced and discussed.

Anomaly detection is used to find outlier abnormal samples in a group of normal samples. Such methods attempt to generate defect-free images with autoencoders (AE) or generative adversarial networks (GANs) to determine an abnormal area by comparing input images and output images. Bergman et al. [12] proposed a modified AE by introducing the SSIM index into the loss function. This enforced the network to

yield a more real image. Mei et al. [13] designed a multiscale denoising encoder based on unsupervised learning to reconstruct normal features in textile images. The reconstruction errors were utilized to realize automatic defect detection. Liu et al. [14] designed a multilevel unsupervised model to reconstruct realistic surface fabric defects by using the generation characteristics of GANs. In [15], a GAN-based anomaly detection technique named AnoGAN was proposed. Some methods based on variational AE were also proposed for anomaly detection, including FAVAE [16], VAE-grad [17], and VE-VAE [18]. In the industrial scenario, these anomaly detection methods based on generating models are limited to the restrictions of training samples. They often have difficulties when excavating deep-level features, and only pay attention to the local pixel-level features, resulting in difficulties in achieving good results.

Some researchers transfer the direction of anomaly detection from the comparison in the image level to the feature level. Cohen et al. [19] first attempted to extract features from pretrained models, and used KNN to compare features. In [20], anomaly detection based on feature reconstruction was proposed. However, it required additional feature fusion modules. Bergman et al. [21] proposed the largest industrial anomaly detection dataset, MVTec, and designed a method based on knowledge distillation for feature comparison. In order to improve the accuracy of anomaly localization, the input image in the model had to be divided into patches, which greatly increased the time and costs.

Different from the current conventional anomaly detection methods, our novel approach mainly focused on anomaly localization (i.e., segmentation) instead of anomaly detection (classification level). Moreover, it put forward the idea of two anomaly comparison modules in the feature space instead of image space. Our method was evaluated on the carpet database, and showed the abilities of unsupervised methods in surface defect detection. The extensive experimental results demonstrated the effectiveness of the proposed method in other industrial applications.

3. The Proposed SFC Framework

The proposed framework SFC had two parts, which are presented in Figure 1. The two parts were the pretrained feature extraction and the feature comparison. In the pretrained feature extraction part, the normal samples without defects were input into the network for feature extraction. Then, the extracted features were employed as the input for the next part. The feature comparison part contained the SRM and SFD. In the training stage, the purpose of these two modules was to model the feature representation of normal samples, ensuring that the input and output of the module were the same. The SRM was an AE-based reconstruction network to reconstruct the input feature. The SFD network was based on a knowledge-distillation network. Its structure was a simplified version of the pretrained network, and was also used to mimic the input features. In the inference stage, the features differed between the input and output of the SRM, representing the anomaly score. Moreover, the features that differed between the pretrained feature extraction and output of the SFD also represented the anomaly score. Both components of the SFC framework were trained simultaneously.

Next, we will introduce various component modules in the model in detail, including feature extraction, SRM, SFD, and the final module for obtaining anomaly scoring, which are presented in Section 3.1, Section 3.2, Section 3.3, and the final Section 3.5. The loss function of the network is described in Section 3.4.
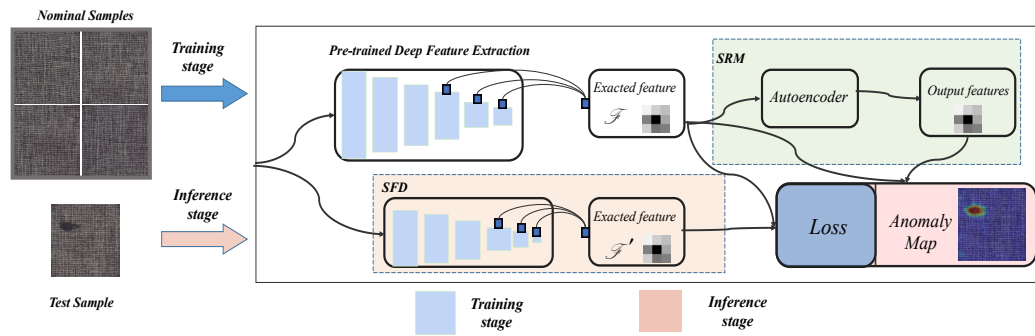
Figure 1. The pipeline of the SFC architecture.

3.1. Pretrained Deep Feature Extraction

In the image-reconstruction-based methods [15–17], the features were trained by the GAN and the encoder in the AE. Due to few training samples being available in industrial scenarios, directly using network training to extract features was not a sensible approach. The image-based pretrained model was a very effective feature extractor, and the features had generalization ability in different scenes. We used unlabeled training data $x = x_k \in X$ sampled from the manifold X of images showing normal anatomy. With n = 1, 2,..., N, we employed a collection of N fabric images that exclusively depicted normal anatomy, where $X_n \in R^{u \times v}$ is an intensity image of size u × v. We defined the feature extractor as F. When a training image x_k is input, the corresponding extracted features can be expressed as:

$$\mathscr{F} = F(x_k) \tag{1}$$

The resolution of the obtaining features $\mathscr{F}$ is h×w×d, and d represents the dimension on the feature channel. The feature maps were processed by extensive convolutional and nonlinear operations. As the pretrained weights were fixed, the h×w×d dimension features could be quickly obtained, and no additional network training was required. In the pretrained network, the size of each critical layer F_i extracted was inconsistent. To ensure that the features extracted from each critical layer F_i could be fused, it was often necessary to use the resize operation. Generally, the feature maps after the pooling operation were selected as $\mathscr{F}_i$. The final $\mathscr{F}_i$ was generated by upsampling and concentrated into the final feature tensor $\mathscr{F}$. Moreover, each position in the original input image x_k corresponded to the position in $\mathscr{F}(i,j)$ after feature extraction. There was only a scaling relationship, which was convenient for exact anomaly localization.

3.2. Self-Feature Reconstruction Module

Motivated by the image-reconstruction method [18,20], we trained a feature reconstruction to automatically reconstruct $\mathscr{F}$. $\mathscr{F}$ represented a feature distribution that was used to describe the distribution of normal features in defect-free sample X. During AE training, the network was optimized for the difference between the input and output. Given input $\mathscr{F}$ from feature space $\in R^{h \times w \times d}$ of dimension d, the auto-encoder was employed to produce features that were as close as possible to the input. In this step, the SRM attempted to reconstruct features of the normal sample distribution.

The detailed network structure in the SRM is shown in Figure 2. In order to ensure that the size information of the image was not lost, 1 × 1 convolution was utilized in the network. The encoding unit in the SRM contained three convolution layers that were compressed in the bottleneck layer. The size of the input feature was constant, but the number of channels gradually compressed from d to d_3. In the decoding unit, three convolution layers were also employed, and the number of channels of the three convolution layers corresponded to the encoding unit. This ensured that the SRM could restore input features $\mathscr{F}$.
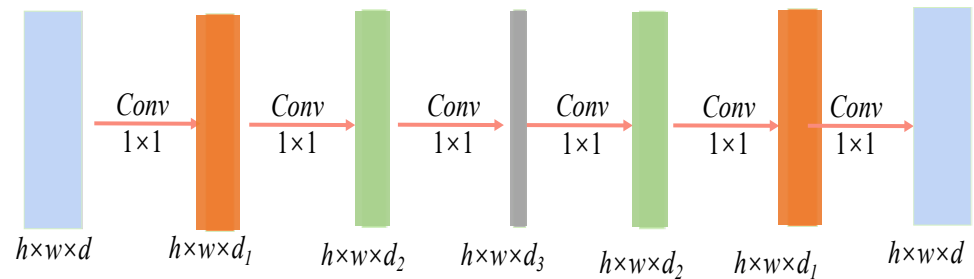
Figure 2. The architecture of the SRM.

3.3. Self-Feature Distillation

SFD aims to train a student network, *S*, that uses a pretrained network to locate anomalies in images. It is trained to mimic the comprehensive behavior of a teacher network *T*. In order to reduce the network parameters, most of the early knowledge distillation methods migrate the information in the complex network to the network with smaller parameters, and ensure its performance is unchanged. These methods are essentially based on the consistent output of large network *T* and small network *S*. In the SFD, the intermediate feature of $\mathscr{F}$ on the normal training data is transferred to the student network instead of the final output.

Since $\mathscr{F}$ was obtained from several critical blocks, the SFD promoted the student network to gain $\mathscr{F}$'s knowledge on normal samples by conforming its intermediate representations in layers to $\mathscr{F}$'s representations. Hence, *i* represented the *i*-th intermediate layer in the networks, and the source features of that intermediate layer as $\mathscr{F}_i$ and the cloner's features as S_i. The feature maps after the pooling operation were selected as S_i. The final *S* was generated by concentrating the branch feature S_i. Our knowledge distillation concept was not to ensure that the final outputs of *T* and *S* were consistent, but to make their intermediate feature outputs consistent.

In this paper, VGG19 was selected for the pretrained feature extraction network. As can be seen in Figure 1, the three feature maps from different layers constituted a final extracted feature $\mathscr{F}'$. The features of different layers represented different semantic information. Compared to the front layer, the semantic information of the rear layer was more abundant. The structure of the SFD is shown in Figure 3.
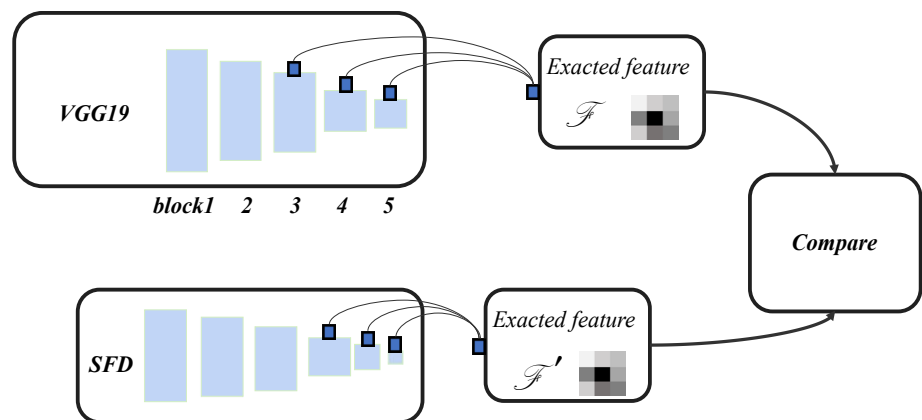

Figure 3. The architecture of the SFD.

In order to better simulate the output of *T*, the structure of *S* was not consistent with *T*. *S* used a simplified convolutional neural network. It only needed to ensure that the output of the key intermediate feature layer was consistent with the output

of the teacher network. As the teacher network was VGG19, we selected the three convolutional block layers to conduct the knowledge distillation. The names of the three layers in the VGG19 were block3_conv4, block4_conv4, and block5_conv4. As shown in Figure 3, each layer feature was upsampled and then concentrated into the final feature tensor. The detailed structure of the student network is shown in Table 1.

Table 1. The structure of the student network in the SFD.

Layer	Number of Filters	Kernel Size	Stride
Input	-	-	-
Conv1	64	3×3	1
Conv2	64	3×3	1
Max pool1	-	-	2
Conv3	64	3×3	1
Conv4	256	3×3	1
Max pool2	-	-	2
Conv5	64	3×3	1
Conv6	64	3×3	1
Conv7	512	3×3	1
Max pool3	-	-	2
Conv8	64	3×3	1
Conv9	64	3×3	1
Conv10	512	3×3	1

BatchNormalization was used after each convolution layer for convergence stability. Three layers in the student net—Max pool2, Max pool3, and Conv10—were used for matching the teacher networks. The parameters of T came from the pretraining model, and could remain unchanged. In the training process of the SFD, the parameters of S were optimized.

3.4. Training Loss

Two losses were used to optimize the proposed network. L_{srm} and L_{sfd} were each aspect from the SRM and SFD. The first, L_{srm}, attempted to optimize the similarity between $\mathscr{F}$ and its reconstruction feature $\mathscr{F}_r$. The loss function for guiding the training of SRM is:

$$L_{srm} = \sum_{i}^{N} \; || \mathscr{F} \; (x_i) \; - \mathscr{F}_r \; (x_i) \; ||_2 \tag{2}$$

where $|| \cdot ||$ is the L$_2$ norm, and x_i is the input image.

The SFD was used to optimize the composite intermediate layers of T and S, and ensure that the outputs of S and T were as close as possible. Hence, L_{sfd} is expressed as:

$$L_{sfd} = \sum_{i}^{N} \; || \mathscr{F} - \mathscr{F}' ||_2 \tag{3}$$

By considering the two parts losses, L_{total} is formulated as:

$$L_{total} = L_{srm} + \lambda L_{sfd} \tag{4}$$

where λ in the formula adjusts the contributions of the different parts. In Section 4.4, we will analyze the influence of different λ values on the anomaly segmentation effect in the dataset.

3.5. Detection of Fabric Anomalies

After the training phase, the SFC had learned the ability to map features of normal samples into themselves in the SRM. Moreover, the SFD also had the ability to reproduce

the features of normal samples. In the test phase, the abnormal score could be obtained by calculating the affinity between the $\mathscr{F}$ and its reconstruction or distillation results. The anomaly scoring could be obtained from the reconstruction branch and distillation branch. The final anomaly heatmap was produced by the fusion of two anomaly scores:

$$S(x_i) = \beta_1 UP(||\mathscr{F}-\mathscr{F}_r||_2) + \beta_2 UP(||\mathscr{F}-\mathscr{F}'||_2) \tag{5}$$

where x_i is the test sample; β_1 and β_2 represent the influence coefficient of different branches on the final anomaly score; and UP(.) is the upsampling process, which employed a bilinear interpolation operation to enlarge the difference to the size of the input.

4. Experiments

To evaluate the performance of the proposed method, several sets of experiments were conducted. In this section, the experimental datasets and implementation details are given first. Then, the overall performance of our method is compared with several state-of-the-art models. Third, the ablation analysis of the model is discussed. Finally, we deploy the model in a real industrial environment for practical usage.

In our experiments, the classical fabric carpet dataset in MVTec AD was utilized. The carpet dataset is a real-world dataset composed of five different industrial defect types, and defects were manually generated that occur in real-world industrial inspection scenarios. It has been widely used for performance verification. The training set contained 280 defect-free images. 89 defective images of five defect types (color, cut, hole, metal_contamination, and thread), and 28 defect-free images were included in the testing sets. An image of an example for each defect type is shown in Figure 4. For quantitative comparison of the performances of the different methods, a threshold-independent evaluation indicator, the pixel-level area under the receiver operating characteristic curve (AUROC), was adopted; this indicator is widely used for performance evaluation of anomaly localization. All the experiments were conducted with the deep-learning toolbox Keras with an NVIDIA 3090 GPU. The main architecture of the SFC in the experiment is shown in Figure 1. All input images were scaled to 256 × 256 pixels.
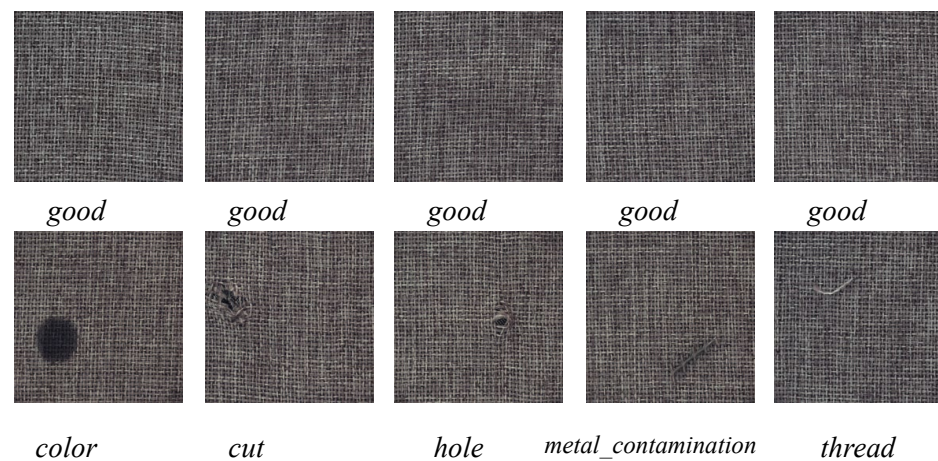

Figure 4. Examples of good and defective images from the carpet dataset.

4.1. Overall Performance Comparison

To verify the effect of the proposed method more fully, the segmentation performance of the proposed method was compared with several state-of-the-art methods, including autoencoders (AE-SSIM [12] and multiscale AE [13]), a generative adversarial network (AnoGAN [15]), variational methods (FAVAE [16], VAE-GRAD [17], and VE-VAE [18]) and other superior unsupervised algorithms (SPADE [19], DFR [20], and US [21]). The results of the methods [12,13,15–21] were taken from the literature.

The segmentation results of these methods on the carpet dataset are exhibited in Table 2. The table reports the segmentation results, which showed that margins of 0.80% and 1.30% were achieved compared with the second-best and third-best methods. The proposed method handled the fabric textured surfaces well, demonstrating the superiority and stability of our method and its potential to be a unified model for defect inspection in industrial applications.

Table 2. Results for the carpet dataset (pixel AUROC).

Method	Pixel AUROC
AE-SSIM [12]	87.0
Multiscale AE [13]	89.0
AnoGAN [15]	54.0
FAVAE [16]	96.0
VAE-grad [17]	74.0
VE-VAE [18]	78.0
SPADE [19]	97.5
DFR [20]	97.0
US [21]	93.5
Ours	98.3

4.2. Visual Inspection Result

The visual inspection results of our proposed method using the carpet dataset are shown in Figure 5. The proposed method could locate the anomaly accurately on five classical defects. It can be seen that the thin texture changes could be captured well by our model. This indicated that our approach has application prospects in real scenarios.
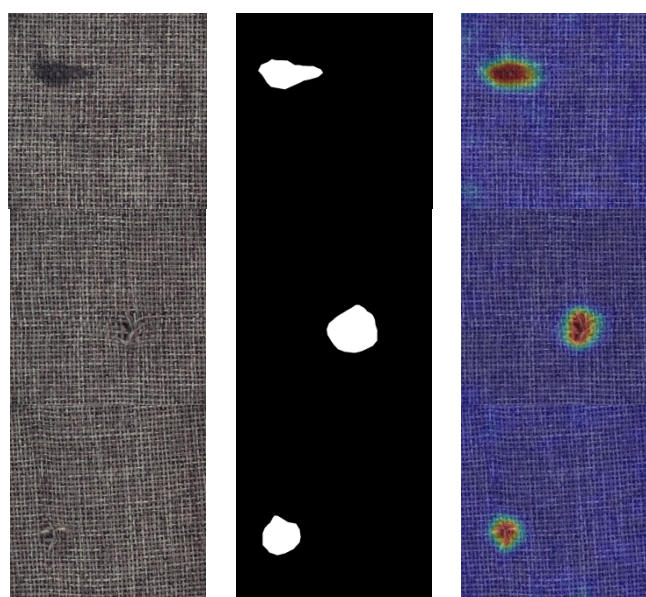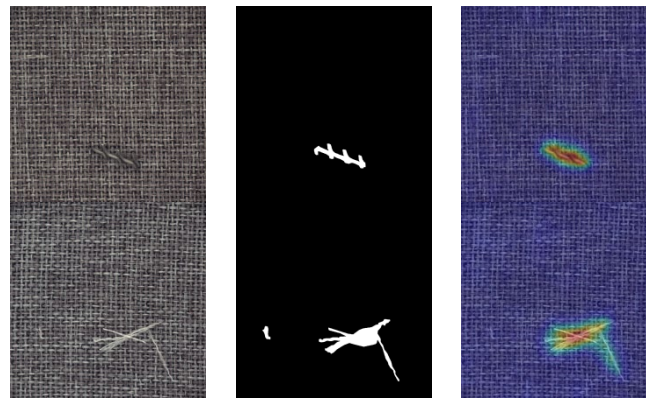
Figure 5. Inspection results of our proposed method using the carpet dataset. The first row to the last row represent the color, cut, hole, metal_contamination, and thread defects, respectively. The first column is the original image, the second column is the ground truth, and the third column is an anomaly map.

4.3. Influence of a Single Module

Previous studies showed that the single reconstruction module or distillation module had the fine discrimination ability of anomaly segmentation. The proposed method experimentally explored the effect of combination modules for fine industrial anomaly segmentation. Table 3 shows the influence of a single module on anomaly localization.

Table 3. Influence of a single module on the SFC.

Method	Pixel AUROC
Single SRM	98.1
Single SFD	97.4
Ours	98.3

As shown in Table 3, the single SFD showed the weakest discriminability, while the combination modules had the highest discriminability. However, the effect of the SRM module exceeded that of the SFD. This was because the reconstruction branch was highly discriminative compared to the distillation branch. The method of distillation was only mimicking input features to some extent, but it ignored subtle abnormalities.

4.4. Influence of the Hyperparameter λ

The hyperparameter λ was used to balance the relative contributions of the SRM and SFD in the training phase. The best value was achieved when the data with normal features could be well modeled. To illustrate the influence of λ intuitively, the carpet dataset was employed to perform the verification. Table 4 shows the influence of hyperparameter λ. When the λ increased, the inspection performance of the carpet increased smoothly. This occurred because the loss of the SRM dominated the entire loss. The loss of the SFD can be regarded as an auxiliary loss.

Table 4. The influence of hyperparameter λ in the carpet dataset.

λ	0.001	0.01	0.1	1
Pixel AUROC	98.1	98.1	98.3	98.3

4.5. Setting of the Hyperparameter β_i in Equation (5)

The fusion mechanism aimed to take advantage of the anomaly representation in different feature spaces. The values of β_i, which represented the significance of SRM and SFD, were used to obtain the fusion in Equation (5). This was a critical parameter that affected

the inspection result. We aimed to investigate the sensitivity of the combination of β_i. Table 5 shows the outcomes of various combinations. It can be seen in Table 5 that the fusion experimental effect was generally better than that of a single branch. The experimental results showed that fusion could improve segmentation performance.

Table 5. Results of various β_i combinations in the carpet dataset. "√" means the value of β_i was 1, otherwise it was set to 0.

β_1	β_2	Pixel AUROC
√	-	98.2
-	√	97.6
√	√	98.3

4.6. Applications

To further assess its practical performance, we applied our method to two complicated fabric datasets in real industrial environments established by the AOI equipment.

The star-patterned fabric dataset [22] was published by H.Y.T. Ngan and G.K.H. Pang of the University of Hong Kong, and is composed of 50 images acquired with an AGFA Scan1236 scanner in 2003. The images in the dataset can be regarded as a composite texture of stars according to periodicity. The training dataset contains 25 normal images without defects. At the same time, the test set contains 25 defect images in five categories. The size, shape, and type of defects in different defect images were different, which was quite different from the periodic texture of the background. Anomalies in target images included broken-end, hole, netting-multiple, thick-bar, and thin-bar. All images used in the evaluations are 256 × 256 pixels in gray-value level scale. The details are shown in Table 6.

The color fabric dataset [23] was released by the Tianchi platform [24]. Since the original dataset was used for the object detection task, we produced a new dataset for unsupervised fabric defect localization. It consisted of 168 nondefect samples for training. The test sample included 200 defective samples and 24 defect-free samples. The image size in all datasets was 256 × 256 pixels. The anomalies in the target images were varied, including long-span scratches and small dots. Pixel-level labeled ground-truth images were provided for evaluation. Accurately locating anomalies on a complicated actual texture background was a difficult challenge. The details are shown in Table 7.

Table 6. Star-patterned fabric dataset and inspection results.

Dataset	Training Images	Testing Images					Pixel AUCROC	Time/ms
		Broken-End	Hole	Netting-Multiple	Thick-Bar	Thin-Bar		
Star-patterned fabric	25	5	5	5	5	5	99.0	120

Table 7. Color fabric dataset and inspection results.

Dataset	Training Images	Testing Images		Pixel AUCROC	Time/ms
		Defect	Good		
Color fabric	168	200	24	87.6	200

The anomaly maps of the two datasets are described in Figures 6 and 7, demonstrating the great potential of our method in industrial applications. Our approach was not only suitable for a uniform texture background, but also adapted to defect detection with a challenging nonuniform color background.
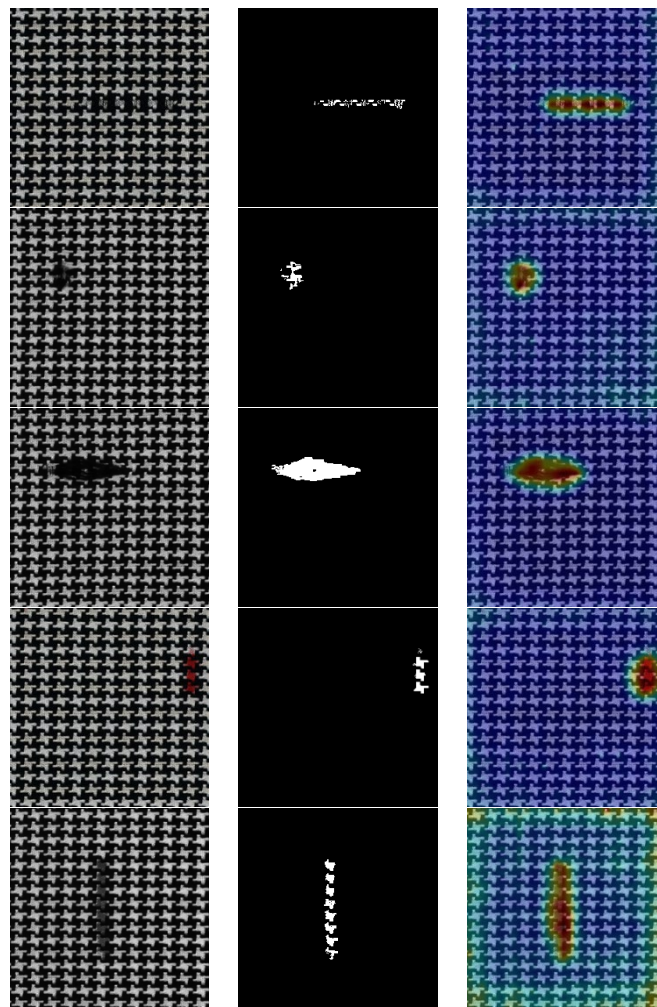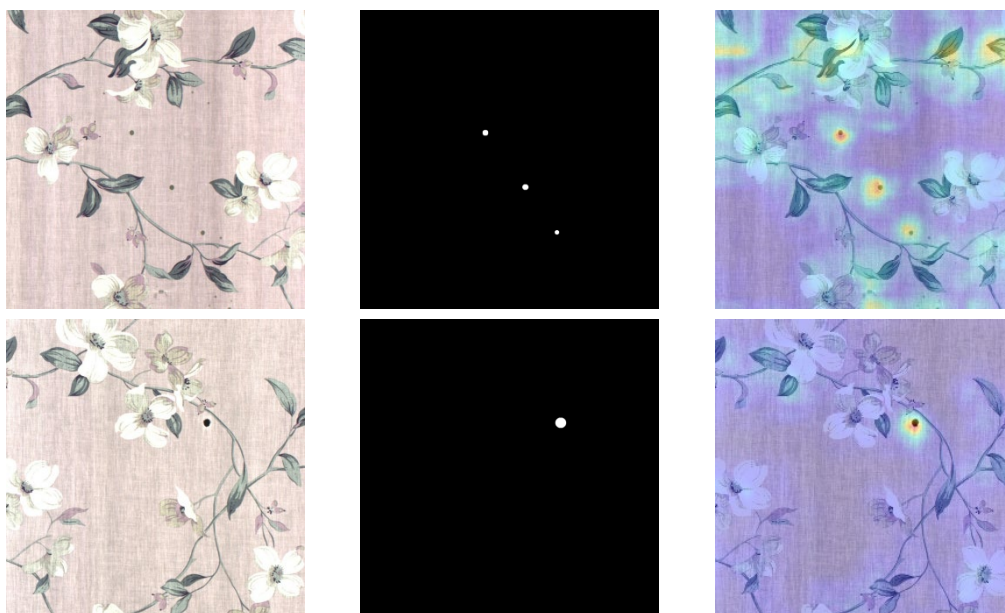
Figure 6. Inspection results of our proposed method on the star-patterned fabric dataset. The first row to the last row represent the broken-end, hole, netting-multiple, thick-bar, and thin-bar defects, respectively. The first column is the original image, the second column is the ground truth, and the third column is an anomaly map.
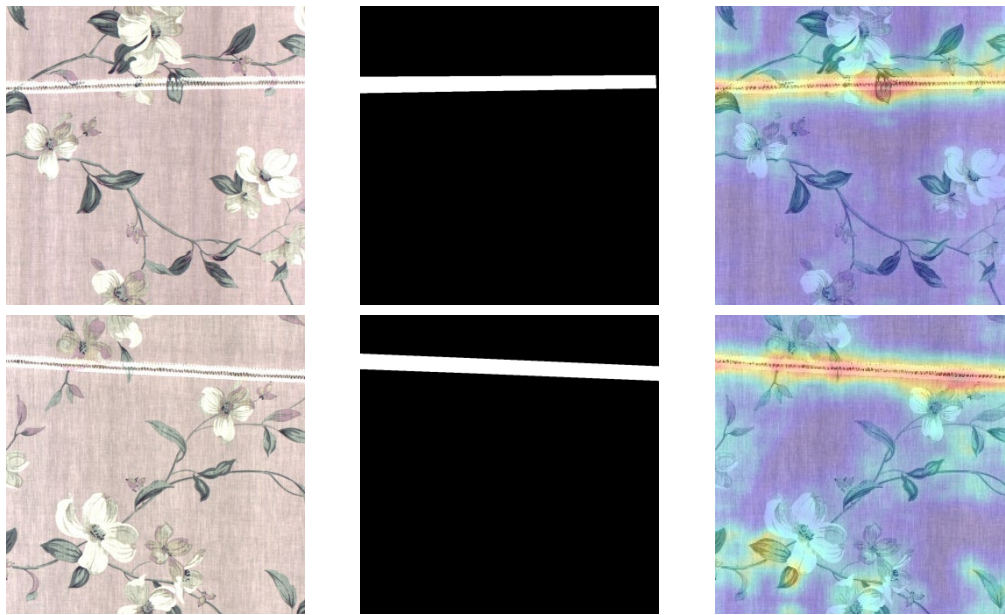
Figure 7. Inspection results of our proposed method on the color fabric dataset. The first two rows represent a tiny dot defect, and the latter two rows represent large-span scratch defects. The first column is the original image, the second column is the ground truth, and the third column is an anomaly map.

In addition, the comparison of different methods (AE-SSIM [12], multiscale AE [13], FAVAE [16], SPADE [19], and US [21]) used on the above two datasets is shown in Figure 8. We compared our method with AE-SSIM [12] and multiscale AE [13] using their publicly official code. Since the official projects are not publicly available, we used third-party implementations for FAVAE [16], SPADE [19], and US [21].
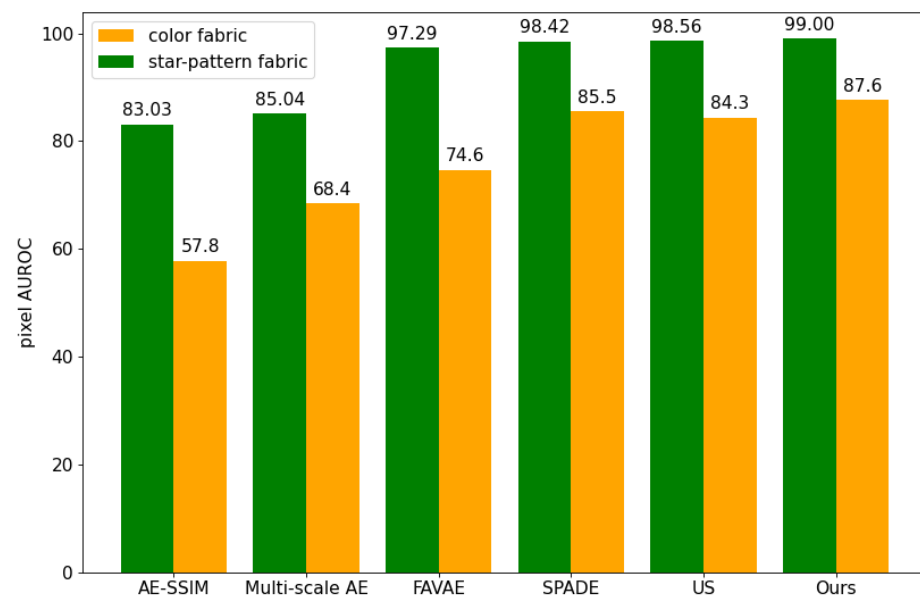

Figure 8. The comparison of different methods used on the two datasets.

As for the star-patterned fabric dataset, three methods, including FAVAE [16], SPADE [19], and US [21], also achieved a more than 97% pixel AUROC value. In the color fabric dataset, the variations between the images were large, our method achieved the best results. Although the FAVAE [16] could obtain internal defect-free features, when the external imaging difference between defect-free images was large, tiny defects were covered

up in complex textures, resulting in false detection. The pixel AUROC of FAVAE [16] was not over 75%. SPADE [19] and US [21] were lower than our method in the color fabric dataset, at 2.1% and 3.3%, respectively. Overall, the proposed SFC method achieved a comparable result against the state-of-the-art methods.

5. Conclusions

A unified model for texture defect inspection in industrial applications was designed in this paper, and was driven in an unsupervised learning fashion. This method was trained on only a few defect-free samples. It utilized the SFC architecture, which contained the SRM and the SFD. The anomalies were detected via a combined anomaly score based on feature reconstruction and feature distillation. Compared with the traditional methods operating in image space, the comparison of feature space could better locate the anomalies of fiber texture surfaces. Extensive experimental results showed that the proposed method achieved state-of-the-art inspection performance. In the future, we will apply our methods to more industrial surfaces, not just fabric surfaces.

Author Contributions: Conceptualization, Z.P. and S.M.; methodology, X.G. and S.M.; software, X.G. and S.M.; validation, S.M., B.W., and X.X.; formal analysis, X.G. and Z.P.; investigation, X.X. and S.M.; resources, Z.P.; writing—original draft preparation, Z.P., X.G., and S.M.; data curation, X.G.; writing—review and editing, X.G. and S.M. All authors have read and agreed to the published version of the manuscript.

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

References

1. Xie, X. A review of recent advances in surface defect detection using texture analysis techniques. *Electron. Lett. Comput. Vis. Image Anal.* **2008,** *7*, **1**–22.
2. 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.
3. Jing, J.; Wang, Z.; Rätsch, M.;Zhang, H. Mobile-Unet: An efficient convolutional neural network for fabric defect detection. *Text. Res. J.* **2020**, https://doi.org/10.1177/0040517520928604.
4. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
5. Zhu, Z.; Han, G.; Jia, G.; Shu, L. Modified densenet for automatic fabric defect detection with edge computing for minimizing latency. *IEEE Internet Things J.* **2020**, *7*, 9623–9636.
6. Huang, G.; Liu, Z.; Laurens, V.; Weinberger, K. Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
7. Wu, J.; Le, J.; Xiao, Z.; Zhang, F.; Wang, W. Automatic fabric defect detection using a wide-and-light network. *Appl. Intell.* **2021**, *51*, 4945–4961.
8. Wu, Y.; Zhang, X.; Fang, F. Automatic fabric defect detection using cascaded mixed feature pyramid with guided localization. *Sensors* **2020**, *20*, 871.
9. Mahajan, P.M.; Kolhe, S.R.; Patil, P.M. A review of automatic fabric defect detection techniques. *Adv. Comput. Res.* **2009**, *1*, 18–29.
10. Shi, B.; Liang, J.; Di, L.; Chen, C.; Hou, Z. Fabric defect detection via low-rank decomposition with gradient information and structured graph algorithm. *Inform. Sci.* **2021**, *546*, 608–626.
11. Kang, X.; Zhang, E. A universal and adaptive fabric defect detection algorithm based on sparse dictionary learning. *IEEE Access* **2020**, *8*, 221808–221830.
12. Bergmann, P.; Löwe, S.; Fauser, M.; Sattlegger, D.; Steger C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv* **2018**, arXiv:1807.02011.
13. Mei, S.; Wang, Y.; Wen, G. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors* **2018**, *18*, 1064.
14. Liu, J.; Wang, C.; Su, H.; Du, B.; Tao, D. Multistage GAN for fabric defect detection. *IEEE Trans. Image Process.* **2019**, *29*, 3388–3400.
15. Schlegl, T.; Seebock, P.; Waldstein, S.M.; Langs, G.; SchmidtErfurth, U. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Med. Image Anal.* **2019**, *54*, 30−44.

16.   Dehaene, D.; Eline, P. Anomaly localization by modeling perceptual features. *arXiv* **2020**, arXiv:2008.05369.

17.   Dehaene, D.; Frigo, O.; Combrexelle S.; Eline, P. Iterative energy-based projection on a normal data manifold for anomaly localization. *arXiv* **2020**, arXiv:2002.03734.

18.   Liu, W.; Li, R.; Zheng, M.; Karanam, S.; Wu, Z.; Bhanu, B.; Radke, R.J.; Camps, O. Towards visually explaining variational autoencoders. *arXiv* **2020**, arXiv:1911.07389.

19.   Cohen, N.; Hoshen, Y. Sub-image anomaly detection with deep pyramid correspondences. *arXiv* **2020**, arXiv: 2005.02357.

20.   Yang, J.; Shi, Y.; Qi, Z. DFR: Deep Feature Reconstruction for Unsupervised Anomaly Segmentation. *arXiv* **2020**, arXiv: 2012.07122.

21.   Bergmann, P.; Batzner, K.; Fauser, M.; Sattlegger, D.; Steger, C. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *Int. J. Comput. Vis*. **2021**, *129***,** 1038–1059.

22.   Ngan, H.Y.T.; Pang, G.K.H. Regularity analysis for patterned texture inspection. *IEEE Trans. Autom. Sci. Eng.* **2008**, *6*,131–144.

23.   Frabic Dataset. 2019. Available online: https://tianchi.aliyun.com/competition/entrance/231748/introduction?spm=5176.12281925.0.0.158c7137kWTDaD (accessed on 3 May 2021).

24.   Tianchi Platform. 2019. Available online: https://tianchi.aliyun.com/ (accessed on 3 May 2021).