

Article

Effective Fabric Defect Detection Model for High-Resolution Images

Long Li ¹, Qi Li ^{2,*}, Zhiyuan Liu ² and Lin Xue ² 

¹ Department of Engineering Mechanics, Shanghai Jiaotong University, Shanghai 200240, China; long.li@sjtu.edu.cn

² School of Mechanical Engineering, Dalian University of Technology, Dalian 116081, China; gaven_yangyd@163.com (Z.L.)

* Correspondence: lqdut123@163.com

Featured Application: The research results can quickly and accurately detect defects in the fabric production process.

Abstract: The generation of defects during fabric production impacts fabric quality, and fabric production factories can greatly benefit from the automatic detection of fabric defects. Although object detection algorithms based on convolutional neural networks can be quickly developed, fabric defect detection encounters several problems. Accordingly, a fabric defect detection model based on Cascade R-CNN was proposed in this study. We also proposed block recognition and detection box merging algorithms to achieve complete defect detection in high-resolution images. We implemented a Switchable Atrous Convolution layer to enhance the feature extraction ability of ResNet-50 and upgraded the Feature Pyramid Network to improve the detection accuracy of small defects. Experimental results demonstrated that the proposed model can efficiently perform defect detection in fabric.

Keywords: fabric defect detection; Cascade R-CNN; Feature Pyramid Network; Switchable Atrous Convolution



Citation: Li, L.; Li, Q.; Liu, Z.; Xue, L. Effective Fabric Defect Detection Model for High-Resolution Images. *Appl. Sci.* **2023**, *13*, 10500. <https://doi.org/10.3390/app131810500>

Academic Editors: Donato Cascio and Krzysztof Koszela

Received: 23 May 2023

Revised: 21 August 2023

Accepted: 8 September 2023

Published: 20 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The evolution of digital technologies is placing companies in front of a potential paradigm shift, characterized by greater interconnection and cooperation between systems, people, and information. This technological mix of automation, information, connection, and programming is leading to the birth of the fourth industrial revolution, also known as Industry 4.0. In the field of smart manufacturing, one of the most important sectors is manufacturing inspection, which consists of detecting product defects. Current inspection systems cannot guarantee good performances while keeping processing efficiency and reducing inter-subject variability. Therefore, quality control procedures with both performance and efficiency are essential [1].

Factories produce a large amount of fabric with defects every year, which warrants the development of fabric defect detection methods to ensure its quality. Currently, most factories employ workers to inspect fabric defects, but they are restricted by limitations such as low accuracy and easy fatigue [2]. Hence, a high-precision automatic fabric defect detection system is urgently required. With the development of computer technologies, many scholars have attempted to use image processing algorithms, such as traditional image processing technology and object detection algorithms based on deep learning to automatically identify fabric defects [3]. Traditional image processing technology is used to detect defects with simple fabric pattern backgrounds; it cannot extract features with complex backgrounds. In the context of Industry 4.0, the increasing availability of data, the advancements in computing power, and breakthroughs in algorithm development have

led machine learning (ML) and deep learning (DL) methodologies to develop appealing solutions in different industrial areas such as predictive maintenance, decision support system (DSS), and quality control (QC) [4]. The object detection algorithm based on deep learning has many advantages, such as high robustness and detection accuracy. Consequently, deep learning algorithms have become the preferred choice for fabric defect detection [5,6].

Object detection algorithms based on deep learning can be developed quickly [7]; therefore, several researchers have attempted to use them to automatically detect fabric defects to increase production profit. For example, Hao Zhou et al. proposed a fabric defect detection algorithm based on Faster R-CNN [8]; they mainly used the deformable convolution network and distance IoU loss function to enhance the detection results, but this method is strict in the selection of threshold value and has poor robustness. Based on convolutional neural networks, Peiran Peng et al. proposed a new network for detecting fabric defects [9]; furthermore, they proposed a method to generate priori anchors to locate objects more precisely, this method has achieved good results in fabric images with simple backgrounds, but poor results in fabric images with complex backgrounds. Based on YOLOv3, Junfeng Jing et al. proposed a method to select the number and size of prior frames, followed by adjusting the YOLO detection layer to decrease the error detection rate [10]. The detection speed of this method is fast, but there is still a lot of room for improvement in detection precision. It should be noted that research on the detection of fabric defects has progressed significantly, and the detection of fabric images with small sizes and simple backgrounds has basically reached the industrial demand. Nevertheless, it is difficult to obtain high detection precision on fabric images with a large size and complex background, which is the main focus of this study.

In our work [11], classical object detection models such as SSD [12], YOLO [13–15], Faster CNN [16–18], and Cascade R-CNN [19] were trained and tested. The Cascade R-CNN model was eventually selected as the base network; accordingly, some optimizing strategies were introduced to improve its detection accuracy. Furthermore, we proposed a block recognition algorithm as well as a detection box merging algorithm to detect small defects in high-resolution images. Subsequently, we employed a multi-morphology data augmentation method to improve the detection precision in the case of a small number of defects. Moreover, Switchable Atrous Convolution (SAC) layers were used to replace 3×3 convolutional layers in Stage 2 of ResNet-50 to enhance the feature extraction ability of the backbone network for multiscale targets. In addition, considering the importance of the Feature Pyramid Network (FPN) module in the detection results of multiscale targets, we optimized its up-sampling algorithm and network structure to improve the detection precision for small targets.

The remainder of this paper is organized as follows. In Section 2, we introduce the proposed method. In Section 3, the experimental results are discussed. Finally, Section 4 summarizes this study and provides suggestions for further improvement.

2. Proposed Model Based on Cascade R-CNN

In this section, we introduce the proposed model based on Cascade R-CNN. In the baseline model, ResNet-50 [20] and the FPN model [21] were selected as the backbone and neck network, respectively, along with RoIAlign [22] and cross-entropy loss. To improve the detection accuracy of the defect detection model, we propose a block recognition algorithm and multi-morphology data augmentation method, along with the SAC layer and Content-Aware ReAssembly of Features (CARAFE) model.

2.1. Cascade R-CNN

Faster Region-based Convolutional Neural Network (Faster R-CNN) is a commonly used two-stage detection network [17]; however, this network has limitations. During model training, the Region Proposal Network (RPN) generates many candidate boxes, which are identified as positive samples based on the Intersection over Union (IoU) between

the candidate and true boxes. However, it is difficult to select an appropriate threshold for IoU; when the threshold is significantly small, some candidate boxes that are not the detection targets may get misidentified as positive samples, whereas when the threshold is large, many small target candidate boxes may be omitted, resulting in positive and negative samples imbalance owing to a small number of positive samples. Therefore, it is necessary to design a more robust candidate box screening method.

The overall architecture of Cascade R-CNN is shown in Figure 1. INPUT indicates the input image, Conv indicates the backbone, NECK indicates the neck network, Pool indicates region-wise feature extraction, H(1,2,3) indicates the region of interest heads, B(1,2,3) indicates bounding box regression, C(1,2,3) indicates classification, and B0 indicates proposals in RPN. Cascade R-CNN can solve the problem of threshold selection in a region of interest (ROI) by applying a multistage Region-CNN structure. The network model sets multiple detection heads, as shown in Figure 1, and the output boxes of each stage can be used as input candidate boxes for the next stage. In this study, different thresholds were set for each detection heading simultaneously, and they were gradually increased so that the proposals from the previous stage could adjust to the next step with a higher threshold, thus enhancing the detection accuracy. The thresholds here are hyperparameters that need to be adjusted for different types of data sets.

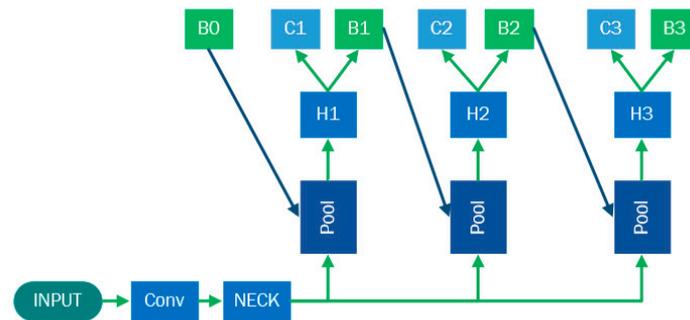


Figure 1. Overall structure of Cascade R-CNN.

2.2. Block Recognition Algorithm

The statistics of fabric defect datasets state that the difference in defect sizes is usually large, and numerous small defects exist. The detection of small targets in high-resolution images is a difficult task in deep learning. If the high-resolution image is resized directly for training, the pixels with small- and medium-size defects in the image will be seriously lost, resulting in the inability of the model to learn its features. Thus, we proposed a defect recognition method called the block recognition algorithm.

In the training process, large images were divided into smaller images, which were then used to train the model. In the inference process, large picture inputs were divided into small images and fed into the network, as shown in Figure 2. Then, the detection results of small images were merged to obtain the final detection results of the large picture input. This novel block recognition method can effectively improve the detection precision of small and medium size defects in high-resolution images.

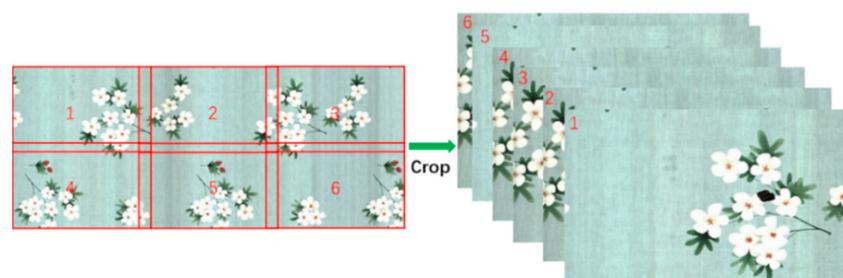


Figure 2. Big images cropped into smaller images.

The algorithm for merging the detection results can be divided into three parts. First, after determining the detection results for each small image, each defect position in the large image was obtained according to the relative position of the large picture and small pictures. We used the Non-Maximum Suppression (NMS) [23] algorithm to remove repeated detection results in the intersection region of small images. Finally, if the IoU [24] of two detection boxes was found to be higher than a threshold, they were merged into a big box. Meanwhile, the largest enclosing rectangle of two small detection boxes was set as the new detection box and the maximum of two scores was set as the new score.

As shown in Figure 3, the two images on the left had overlapping areas, which led to repeated detection. To determine the number of defects and evaluate the quality of the cloth, we located the positions of detected defects in the large image on the right and used the NMS algorithm on the results.

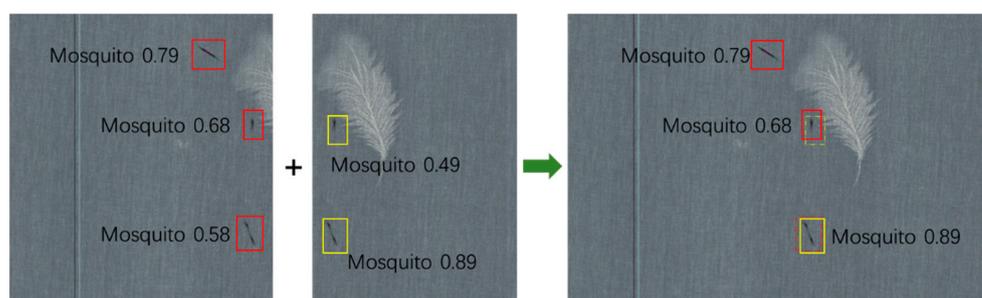


Figure 3. NMS algorithm.

As shown in Figure 4, the two images on the left had overlapping areas, and the defects detected on them were incomplete; however, to accurately evaluate the fabric quality, complete defects must be obtained. The yellow rectangle on the right side of Figure 4 shows the detection result after merging.

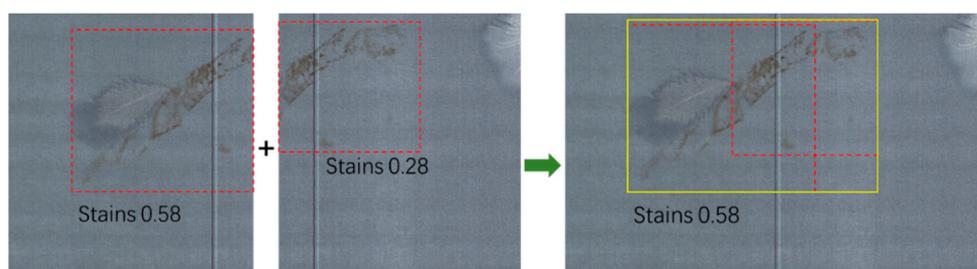


Figure 4. Detection box merging algorithm.

2.3. Multi-Morphology Data Augmentation Method

There are nine kinds of defects in fabric images: stains, sewing_head, holes, shade_yern, broken_figure, mosquito, wrinkles, stop_marks, and thread_ends. The number of defects in nine categories in fabric images significantly varies during fabric defect data collection. Some defects are identified in large numbers, such as stains, shade-yern, and sewing-head, while some are identified in significantly fewer quantities, such as thread-ends, holes, and mosquitos. The model cannot easily learn the features of a smaller number of defects [25]; therefore, we focused on enhancing the small number of defects to improve the quality of the fabric dataset.

Commonly used data augmentation methods such as rotation, mirroring, and shearing have many disadvantages. The generated defects exhibit similar morphology and the number of defects that can be generated is restricted; hence, these methods are not suitable for data augmentation in the case of an extremely small number of defects.

Therefore, we proposed a new data augmentation method to enrich the dataset. Initially, we used a method combining mean filtering and dynamic threshold to extract

fabric defects, with the image background set to white or black. The extracted defects are shown in Figure 5; these images were used in the new data augmentation method.



Figure 5. Extracted defects.

Then, scaling, mirroring, cropping, rotation, morphological processing, and other common data augmentation methods were used to randomly change the morphology of object defects. Finally, we randomly merged the obtained defects into fabric images in batches, as shown in Figure 6. It should be noted that this multi-morphology data augmentation method is not limited by the image background, and can be used to generate defects with different morphologies in batches [26].



Figure 6. Multi-morphology data augmentation.

2.4. Backbone Improvement

2.4.1. ResNet-50 Model

Comparing the commonly used feature extraction networks, we selected Residual Neural Network (ResNet) [20] as the backbone network, considering its detection accuracy and speed. As shown in Figure 7, ResNet-50 mainly comprises five parts, i.e., input layers and four feature extraction stages. The input layer consists of a 7×7 convolutional layer (stride = 2) and a max pooling layer (stride = 2). This layer was used to process the input image; accordingly, the size of the output feature map decreased by four times and its channel size increased to 64. In the four feature extraction stages, each stage except Stage 1 included a down-sampling block. Each residual block consisted of the main path and shortcut connection. The residual structure enabled the construction of deeper networks; moreover, the four feature extraction stages were used to reduce the size of the feature map and obtain deeper semantic information about defects.

In ResNet-50, the number of channels in the output feature map was controlled by the 1×1 convolutional layer, while the feature map size was controlled by the stride of the 3×3 convolutional layer [27]. Figure 7 illustrates the detailed parameters of each convolutional layer and the size of feature maps.

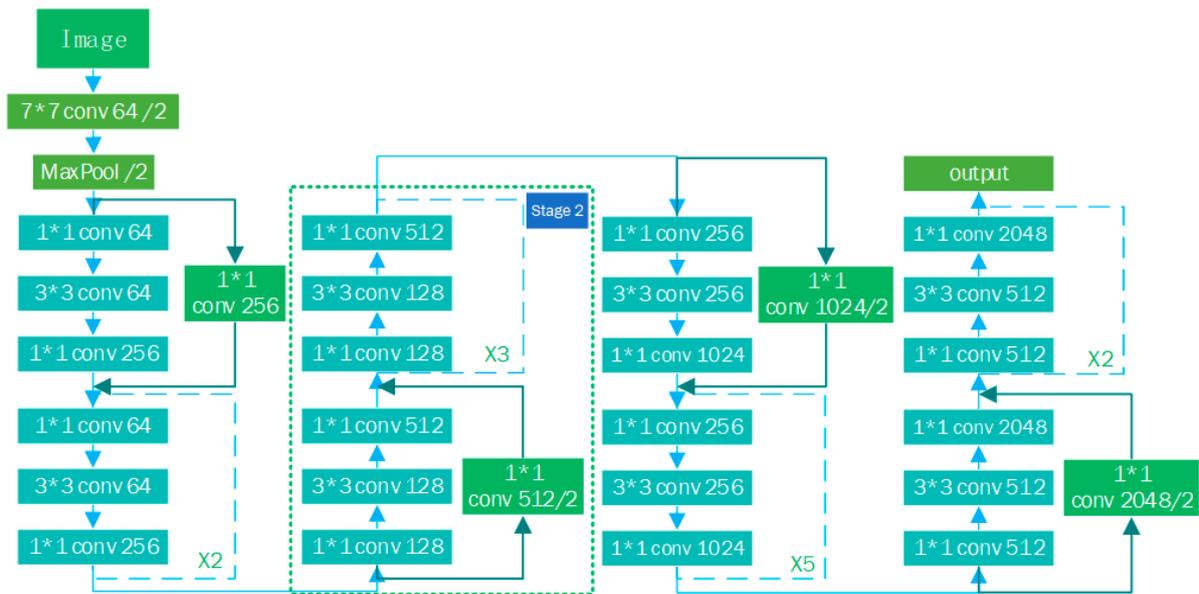


Figure 7. ResNet-50.

2.4.2. SAC Layer

Although ResNet-50 achieved good results on the VOC [28] and COCO [29] datasets, the fabric defect dataset can be further improved. Owing to the large difference in fabric defect size, the introduction of a multiscale module can help in improving the detection accuracy. Consequently, we replaced the 3×3 convolutional layer with SAC (Switchable Atrous Convolution) [30] in Stage 2.

As shown in Figure 8, the SAC module can be divided into three parts, i.e., two global context modules and the main SAC module. The global context modules obtained global information from the global average pooling layer and added it to the feature map. The main SAC module contained three paths, namely the Switch path and two feature extraction paths. The Switch path comprised an average pooling layer with a 5×5 kernel and 1×1 convolutional layer. The two other paths consisted of two convolutional layers with different atrous rates [31].

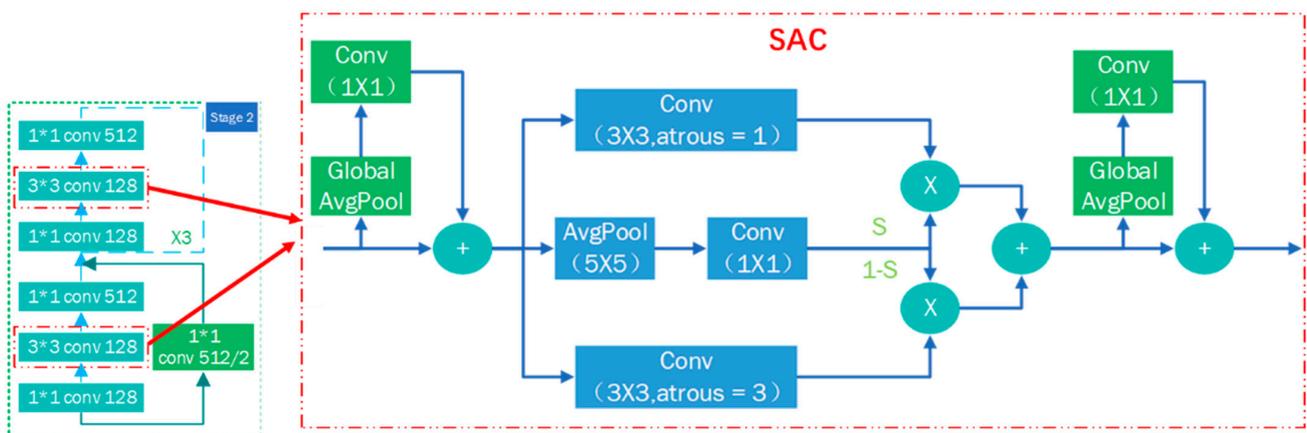


Figure 8. SAC (Switchable Atrous Convolution).

We use $y = \text{Conv}(x, w, r)$ to denote the convolutional operation with weight w and atrous rate r , which takes x as its input and outputs y . Then, we can convert a convolutional layer to SAC as follows:

$$\text{Conv}(x, w, 1) \rightarrow S(x) \cdot \text{Conv}(x, w, 1) + (1 - S(x)) \cdot \text{Conv}(x, w + \Delta w, r) \quad (1)$$

The convolution results were obtained by fusing the defect information acquired at different atrous rates according to Equation (1). In Equation (1), $S(\cdot)$, $\text{Conv}(x; w; r)$, x , w , and r denote the switch operation, convolution operation, input, weight, and atrous rate, respectively.

The input feature maps were convolved at two different atrous rates to obtain the feature information in different receptive fields. The improved backbone model could adapt to the features of different scales; accordingly, this model was used to improve the detection accuracy of multiscale defect targets.

2.5. Improvement of FPN Model

2.5.1. Reconstructing FPN

By analyzing fabric image training data, we found that the span of defect size was considerably large, and some were extremely small or large. Moreover, the feature information of small defects could be easily covered in the top feature map, which complicated the detection of small defects. Therefore, the FPN module was added to improve the detection accuracy of small and multiscale defects.

As shown in the red box in Figure 9, the FPN model contained two main pathways, i.e., the top-down and lateral pathways. The lateral pathway included 3×3 and 1×1 convolution layers. It should be noted that 1×1 convolutional layers in a network are used to unify input channels from different stages to the same size, while 3×3 convolutional layers on each merged map are used to reduce the aliasing effect of up-sampling and down-sampling. The top-down pathway and lateral connections are designed to fuse high-level semantic information to low-level maps.

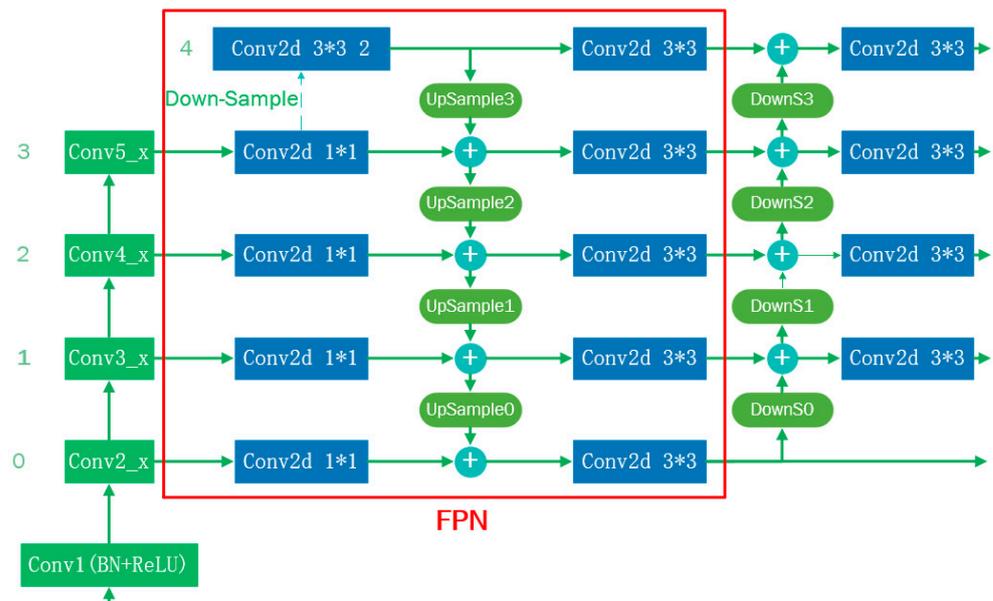


Figure 9. Path augmentation FPN.

It is well known that low-level feature maps contain more accurate location information, while high-level feature maps contain more semantic information. To enhance the localization of information at higher levels, a bottom-up path was created in this study to shorten the information path between lower and higher levels, as shown in Figure 10.

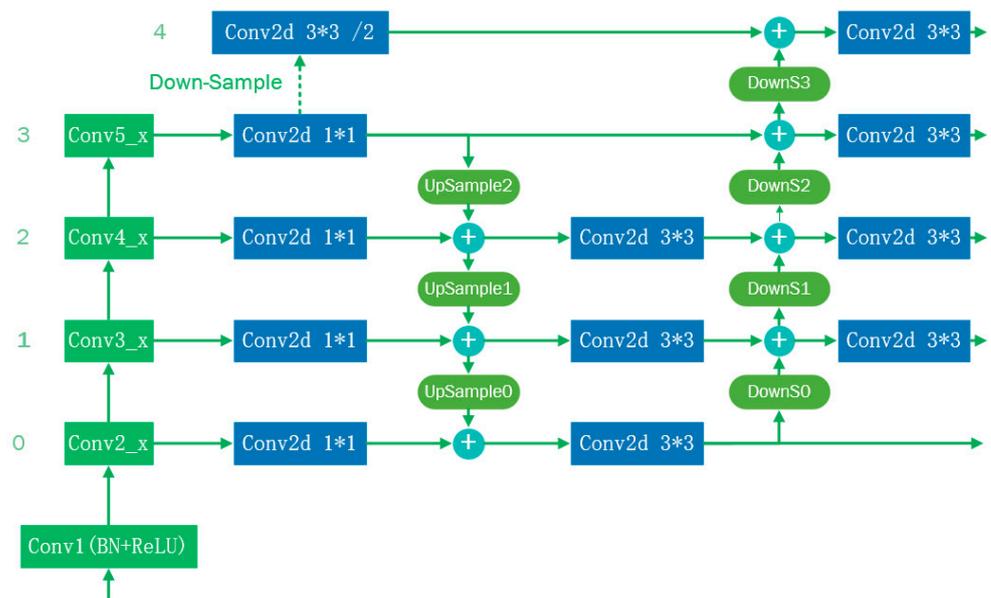


Figure 10. Improved FPN model.

The feature map in level 3 was used repeatedly. It should be noted that this method does not fuse the features of different levels, which is an invalid fusion process. Therefore, we simplified the Path Augmentation FPN [32] module. We removed the feature fusion layer between layers 3 and 4 and deleted the corresponding up-sampling and 3×3 convolution modules. The improved FPN network structure is shown in Figure 10.

2.5.2. CARAFE

Nearest neighbor interpolation and bilinear interpolation are the most frequently used up-sampling operators in the FPN. However, they only consider sub-pixel neighborhoods and cannot utilize the semantic information of feature maps. Therefore, we introduced the CARAFE method [33] to accomplish feature up-sampling.

As shown in Figure 11, CARAFE can be mainly divided into two modules, namely kernel prediction and content-aware reassembly. CARAFE uses the kernel prediction module to predict the up-sampling kernel, which is then used in the content-aware reassembly module to complete the up-sampling operation. In the kernel prediction module, to reduce the amount of calculation, the 1×1 convolutional layer is used to compress channels from C to C_m . Then, the $K_{encoder} \times K_{encoder}$ convolutional layer is used to generate reassembly kernels based on input features, and the output channels are changed to $\sigma 2K_{up}2$. Finally, each reassembly kernel is normalized in the channel direction using a softmax function. In the content-aware reassembly module, the corresponding position of each output pixel in the input feature map is identified and the input position is selected as the center of the $K_{encoder} \times K_{encoder}$ region; accordingly, the output pixel value from the product between the region and up-sampling kernel is obtained.

In comparison to nearest neighbor and bilinear interpolations, CARAFE exhibits many advantages. The up-sampling kernel used to generate output feature maps is related to input semantic information, and CARAFE introduces minimal computation. This up-sampling method can restore the defect information of small defects more accurately, which is beneficial to the recognition of small defects.

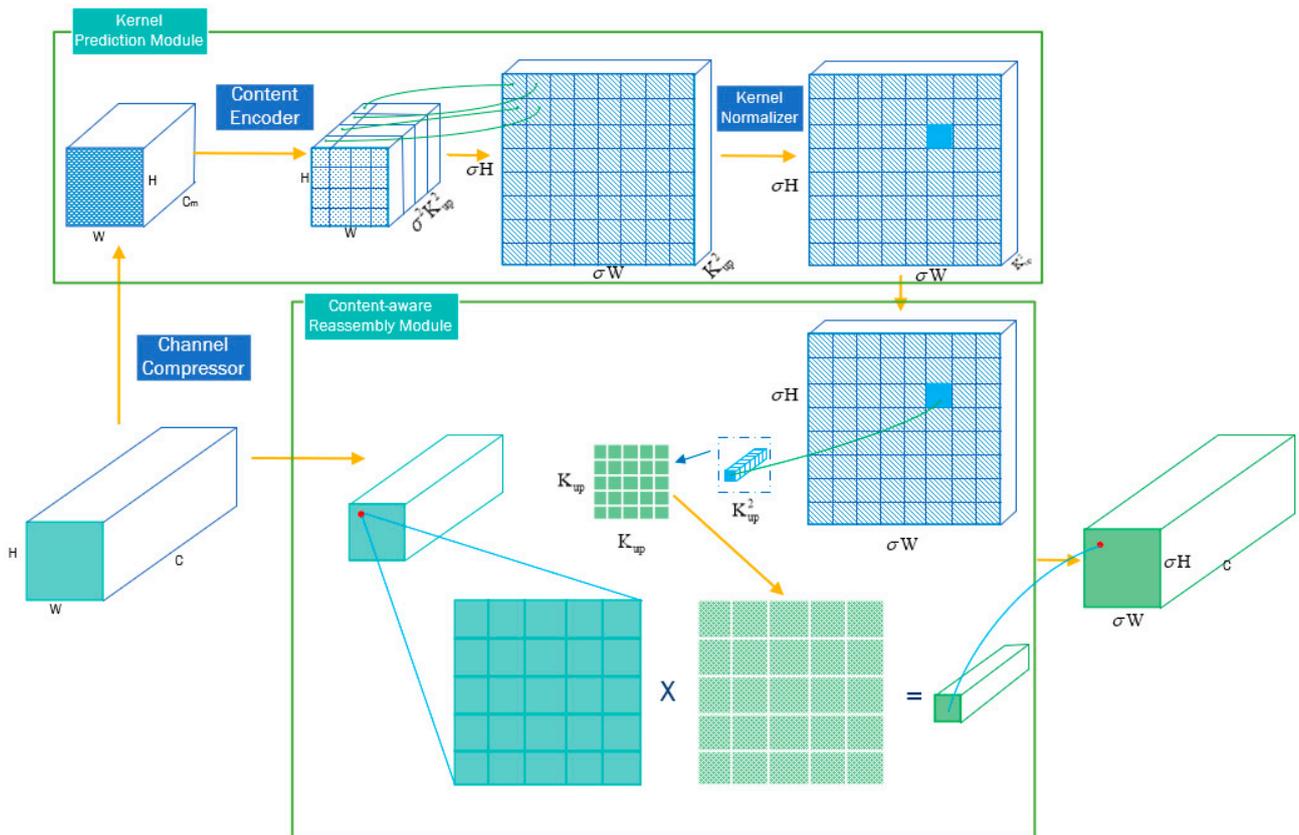


Figure 11. CARAFE model.

3. Experiments

All models in this study were trained in an environment containing an RTX 3080Ti graphics card with 12 GB memory and a 2.9 GHz CPU with 16 GB RAM. During model training, the batch size, learning rate, and epoch were set to 1, 0.00125, and 36, respectively. In this section, we first discuss the dataset and evaluated indicators, followed by details of the experiments.

We performed fabric defect detection based on deep learning by introducing a series of improvements to the baseline model. Table 1 summarizes the detection results of different experiments; the mAP value was observed to be 75.33% eventually.

Table 1. Experimental results.

Experiments	Casade RCNN	IOU Threshold	Boxes Shape	Data Augmentation	Resnet-50 SAC	Carafe	Network Reconstruct	Map
Experiments 1-1	✓	✓						49.44
Experiments 1-2	✓	✓	✓					50.33
Experiments 2	✓	✓	✓	✓				70.33
Experiments 3	✓	✓	✓	✓	✓			71.78
Experiments 4-1	✓	✓	✓	✓	✓	✓		74.67
Experiments 4-2	✓	✓	✓	✓	✓	✓	✓	75.33

3.1. Dataset of Fabric Defect Images

To verify the proposed model, a dataset was constructed [34] with 1604 images; there were 19 different backgrounds and two different sizes (4096×1696 and 4096×1800) of images in the dataset. Based on the different defect shapes and influences on fabric quality, fabric defects were divided into nine categories, namely, stains, sewing_head, holes,

shade_yern, broken_figure, mosquito, wrinkles, stop_marks, and thread_ends, as shown in Figure 12.

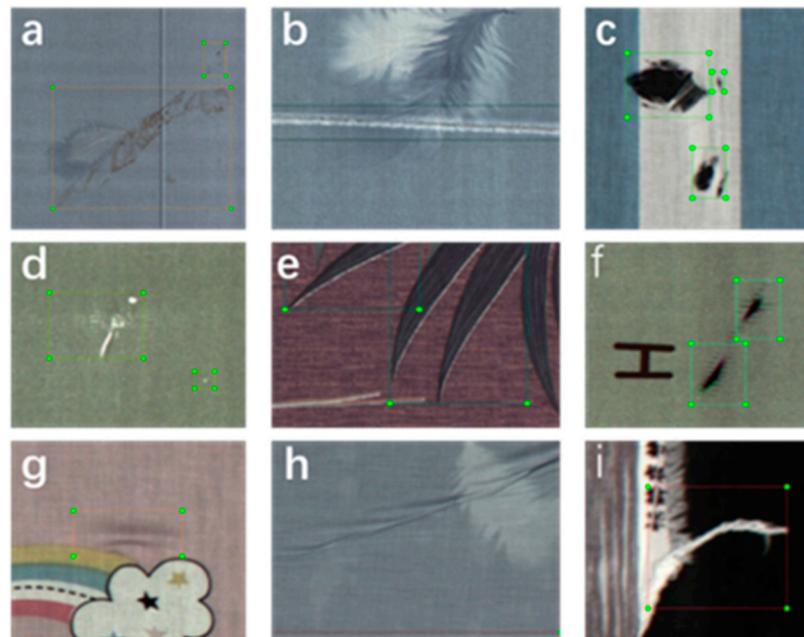


Figure 12. Defects: (a) stains, (b) sewing-head, (c) holes, (d) shade-yern, (e) broken-figure, (f) mosquito, (g) wrinkles, (h) stop-marks, and (i) thread-ends.

To evaluate the proposed model, we adopted stratified sampling, randomly selected 340 images as test data, and ensured that the number of labels for each type of defect in the test data accounted for 20% of the corresponding type of labels, while the remaining 1304 images were used to train and verify the model. As shown in Figure 13, although ResNet-50, the blue, yellow, and gray parts represent the number of defects in the entire dataset, validation and training sets, and the test set, respectively. The training and verification sets were randomly divided during model training in the ratio of approximately 8:2. The main function of the validation set is to test the accuracy rate, recall rate, and mAP that the model can achieve after each round of training, so as to record various indicators of the model after each training.

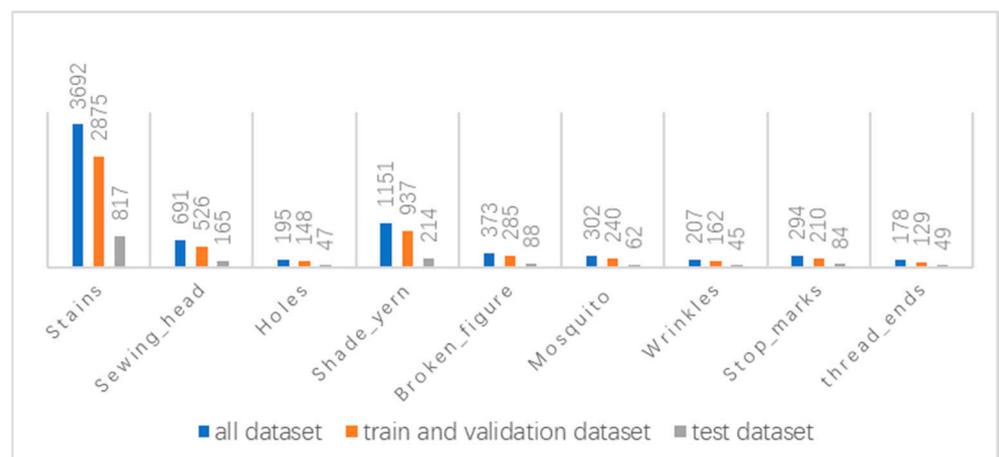


Figure 13. Defects in dataset.

3.2. Evaluated Indicators

In this study, the Mean Average Precision (mAP) was used to evaluate the accuracy of the proposed model; it should be noted that the higher the mAP value, the better the detection performance of the model. In this section, we discuss the calculation of mAP.

In Figure 14, “Precision” and “Recall” represent the ratio of the number of correctly detected defects to all detection results and the total number of defects, respectively.

$$AP = \int_0^1 p(r) dr \quad (2)$$

$$MAP = \frac{\sum_c AP}{m} \quad (3)$$

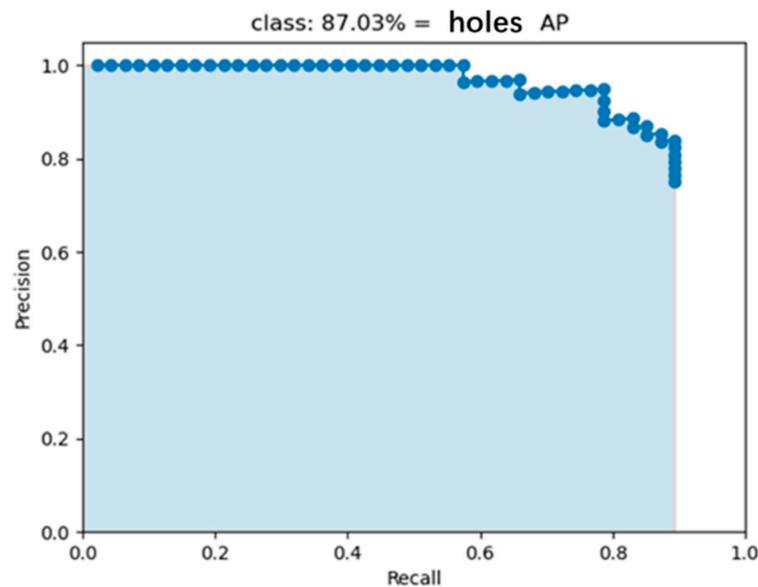


Figure 14. AP curve.

We set different values to verify the detection results and obtained different precision and recall values. As shown in Figure 14, the PR curve was drawn in the coordinates according to the aforementioned different values; the area enclosed by the PR curve represents the AP value. Then, the mAP was calculated according to Equations (2) and (3), where m denotes the number of defect types.

3.3. Experiment I: Block Recognition Algorithm

After obtaining the fabric dataset, we attempted to use high-resolution images to train the proposed model; however, the detection results were extremely poor and the model could not complete the detection task. Thus, we proposed a multiscale defect recognition algorithm. In this study, Cascade R-CNN with the proposed block recognition algorithm was used as the baseline model.

In the experiment, large fabric images were cropped into small images to train the model, and the detection results of small images were merged to obtain those of the large images in the inference stage. Considering the inference time and detection accuracy, a large image was divided into six small images of resolution 1460×960 to train the model.

The IoU threshold and shape similarity of two detection boxes were the two criteria used to determine whether two detection results should be merged. As shown in Figure 15, we only used the IoU threshold as the merge criterion in Experiment 1-1 and included shape similarity as the merge criterion in Experiment 1-2. The mAP value increased from 49.44% to 50.33%, while the AP value of stop_marks defects increased from 37% to 51% after implementing the proposed block recognition algorithm.

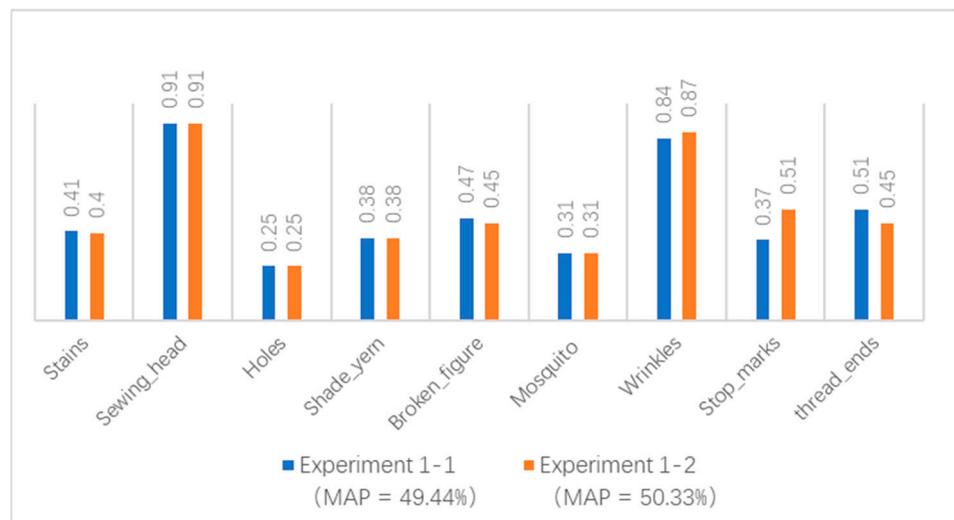


Figure 15. Block recognition algorithm.

3.4. Experiment II: Multi-Morphology Data Augmentation Method

By analyzing the number of defects in Figure 13 and the detection results in Figure 15, we discovered that despite using online data augmentation during model training, the detection results for a small number of defects were poor. Therefore, we proposed a multi-morphology data augmentation method to improve the quality of the fabric dataset.

In Experiment 1-2, we used data augmentation algorithms such as rotation, mirroring, and flipping for all types of defects in the training dataset. In Experiment 2, instead of the previous data augmentation method, we used a multi-morphology data augmentation method for defects such as holes, stains, shade-yern, and mosquitoes to effectively increase the number of defects. The number of added defects is shown in Figure 16.

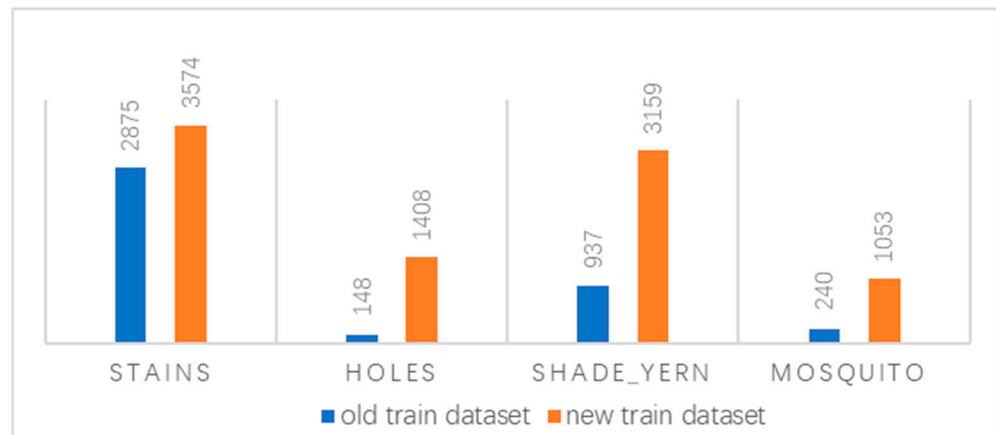


Figure 16. The result of Multi-morphology data augmentation.

As shown in Figure 17, this data augmentation method effectively increased the detection accuracy of a small number of defects. Moreover, the mAP of defects increased by 20.0% and the AP of holes, stains, shade-yern, and mosquitoes with the new data augmentation method rapidly increased.

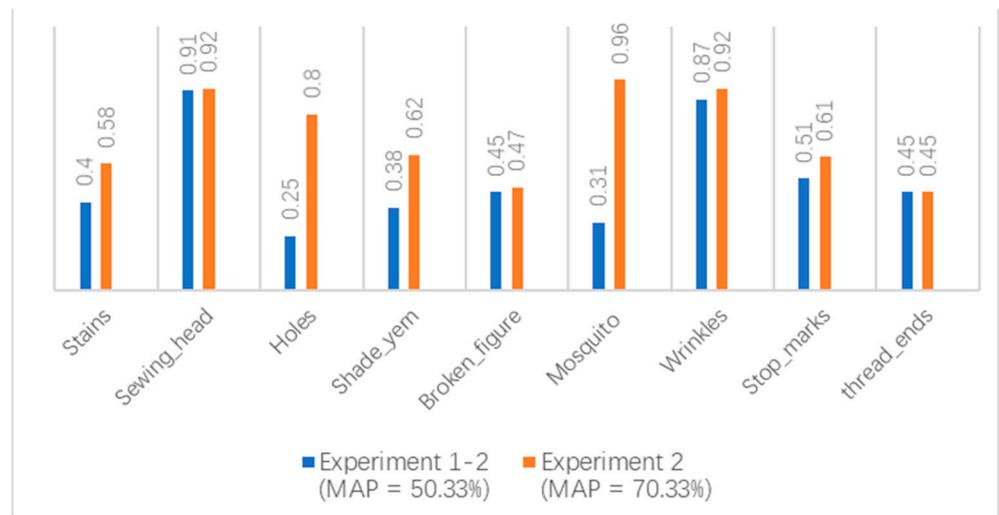


Figure 17. Contrast experiment for data augmentation.

3.5. Experiment III: ResNet Improvement

To verify whether the SAC module could improve the multiscale feature extraction ability of ResNet-50, we set up a contrast experiment. Compared with Experiment 2, we replaced the 3×3 convolutional layer with a SAC layer in Stage 2 of ResNet-50 and compared the detection results of the two experiments. As shown in Figure 18, the AP of stains, broken_figure, and thread_ends evidently improved.

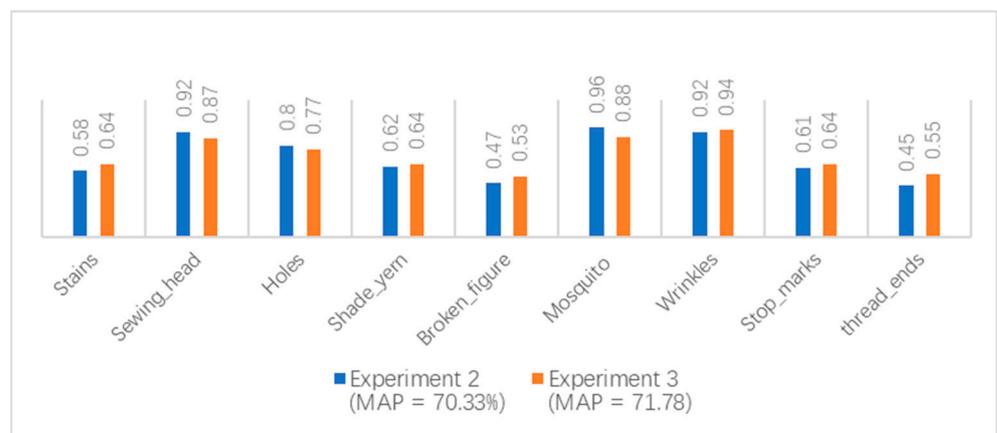


Figure 18. Contrast experiments to improve ResNet.

Experimental results demonstrated that the AP value of some defects decreased, but the detection results of most defects improved. In summary, SAC improved the detection performance of the model, increasing the mAP of defects by 1.45%.

3.6. Experiment IV: Improvement of the FPN Model

To evaluate the performance of the proposed model, we compared the commonly used FPN model with the improved FPN model proposed in the study. We introduced two main improvements in the FPN module, i.e., CARAFE and network path optimization.

The CARAFE module was employed in Experiment 4-1 and the two aforementioned improvements were employed in Experiment 4-2. In Figure 19, the blue, orange, and gray rectangles represent the detection results of Experiments 3, 4-1, and 4-2, respectively.

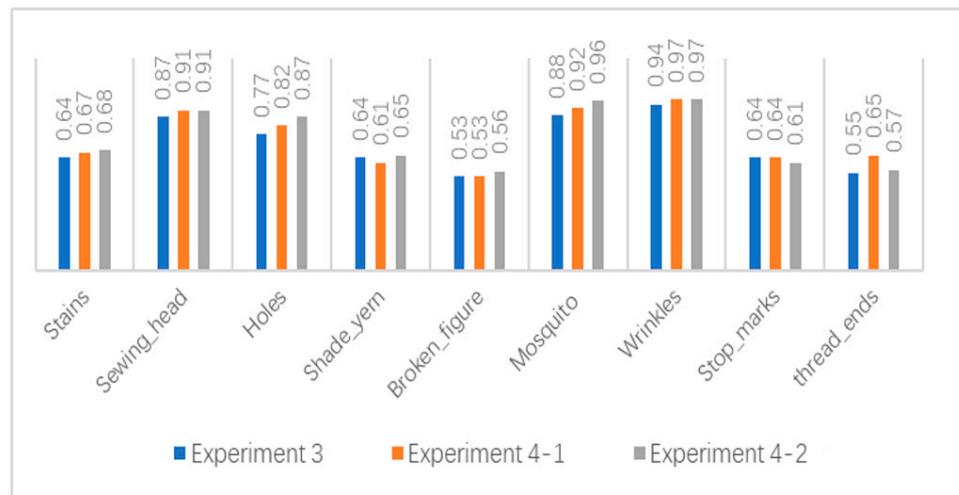


Figure 19. Contrast experiments to improve FPN.

Comparing the detection results of Experiments 3 and 4-1, it was evident that the AP of each defect almost increased and the mAP increased from 71.78% to 74.67%. Then, in comparison to Experiment 4-1, the mAP of defects in Experiment 4-2 increased by 0.66%.

Images in the top part of Figure 20 illustrate the detection results of the original FPN module in Experiment 3, while those in the bottom part show the detection results of Experiment 4-2. The analysis of detection results indicated that the improved FPN module exhibited higher detection and positioning accuracy for small defects. Moreover, the two improvements in the FPN module improved the detection accuracy of defects in fabric.

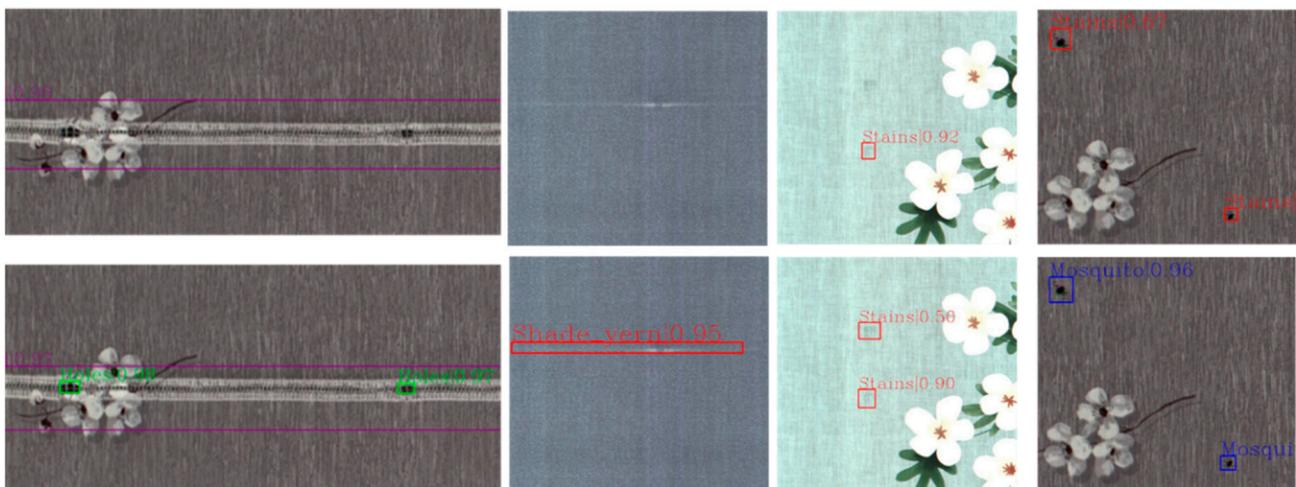


Figure 20. Detection results of experiments.

3.7. Comparison Experiment with Classical Algorithm

To validate the effectiveness of the proposed fabric defect detection model in comparison to other classical target detection algorithms, two groups of contrast experiments were established in this study. All models in these comparison experiments used block recognition and multi-morphology data augmentation algorithms to verify the effectiveness of the proposed model optimization strategy.

Figure 21 illustrates the experimental results; the blue, orange, and gray rectangles represent the detection results of the proposed model, Faster R-CNN, and YOLO V3, respectively. The analysis of the experimental results indicated that the mAP of the proposed model was higher than that of Faster R-CNN and YOLO V3, and our model exhibited higher detection accuracy for small defects such as stains and shade-yarn.

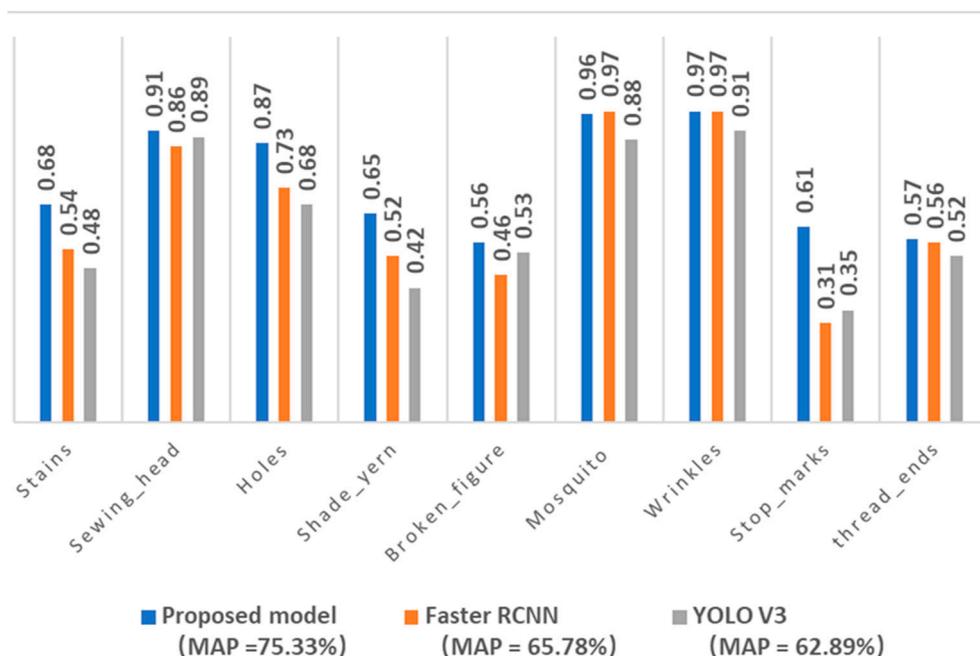


Figure 21. Detection results of comparison algorithm.

4. Conclusions

In this study, we proposed a defect detection model for high-resolution fabric images with complex backgrounds based on Cascade R-CNN. The main findings of our work are discussed below.

The high resolution of fabric images and the large span of defect size lead to extremely poor detection results. Thus, a block recognition algorithm was proposed in this study that yielded a mAP of 50.33%. To solve the problem of a small number of defects resulting in poor detection results, we proposed a multi-morphology data augmentation method to effectively generate defects. This data augmentation method achieved excellent improvement and increased the mAP by 20%. To enhance the feature extraction ability of the backbone network, we replaced the 3×3 convolutional layer in Stage 2 with the SAC module, thus increasing the mAP by 1.45%. Finally, we optimized the up-sampling algorithm and network structure of the FPN module to solve the problem of missing some small defects. The experimental results demonstrated that the detection results for small defects improved, and the mAP increased by 3.55%.

In the actual production process of the fabric, the detection time of a single image is required to be within 0.5 s. The improved model could detect defects in a high-resolution image in 0.37 s, the time for Faster-RCNN to detect a single image is 0.25 s, and the time for YOLO V3 to detect a single image is 0.12 s. Although the detection speed of the improved model is slower than that of the other two mainstream models, it has met the requirements of detection speed. Furthermore, we introduced a series of improvements on the baseline model to perform deep learning-based fabric defect detection. Table 1 summarizes the detection results of different experiments, and the final test image is shown in Figure 22; The mAP value has increased from 49.44% to 75.33%, which is an increase of 52.37%.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

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

References

1. Pazzaglia, G.; Martini, M.; Rosati, R.; Romeo, L.; Frontoni, E. A Deep Learning-Based Approach for Automatic Leather Classification in Industry 4.0. In Proceedings of the Pattern Recognition. ICPR International Workshops and Challenges, Virtual; 10–15 January 2021.
2. Zhao, S.; Wang, J.; Zhang, J.; Bao, J.; Zhong, R. Edge-cloud collaborative fabric defect detection based on industrial internet architecture. In Proceedings of the 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, UK, 20–23 July 2020.
3. Jia, L.; Chen, C.; Liang, J.; Hou, Z. Fabric defect inspection based on lattice segmentation and Gabor filtering. *Neurocomputing* **2017**, *238*, 84–102. [[CrossRef](#)]
4. Rosati, R.; Romeo, L.; Vargas, V.M.; Gutiérrez, P.A.; Hervás-Martínez, C.; Frontoni, E. A novel deep ordinal classification approach for aesthetic quality control classification. *Neural Comput. Appl.* **2022**, *34*, 11625–11639. [[CrossRef](#)]
5. Li, F. Bag of tricks for fabric defect detection based on Cascade R-CNN. *Text. Res. J.* **2020**, *91*, 004051752095522. [[CrossRef](#)]
6. Liu, Z.; Huo, Z.; Li, C.; Dong, Y.; Li, B. DLSE-Net: A robust weakly supervised network for fabric defect detection. *Displays* **2021**, *68*, 102008. [[CrossRef](#)]
7. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
8. Zhou, H.; Jang, B.; Chen, Y.; Troendle, D. Exploring Faster RCNN for Fabric Defect Detection. In Proceedings of the 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), Irvine, CA, USA, 21–23 September 2020; pp. 52–55.
9. Peng, P.; Wang, Y.; Hao, C.; Zhu, Z.; Liu, T.; Zhou, W. Automatic Fabric Defect Detection Method Using PRAN-Net. *Appl. Sci.* **2020**, *10*, 8434. [[CrossRef](#)]
10. Jing, J.; Zhuo, D.; Zhang, H.; Liang, Y.; Zheng, M. Fabric defect detection using the improved YOLOv3 model. *J. Eng. Fibers Fabr.* **2020**, *15*, 155892502090826. [[CrossRef](#)]
11. Xue, L.; Li, Q.; Lu, Y.; Zhang, D.; He, Q.; Wang, H. Fabric Defect Detection Based on the Improved Cascade R-CNN. *Acad. J. Comput. Inf. Sci. AJCIS* **2021**, *4*, 81–87.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. *SSD: Single Shot MultiBox Detector*; Springer: Cham, Switzerland, 2016.
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
14. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 6517–6525.
15. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
18. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
19. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
21. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Chengdu, China, 2017.

22. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
23. Neubeck, A.; Gool, L. Efficient Non-Maximum Suppression. In Proceedings of the International Conference on Pattern Recognition, Hong Kong, China, 20–24 August 2006.
24. Rahman, M.A.; Yang, W. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 12–14 December 2016; Springer: Cham, Switzerland, 2016.
25. Luo, Y.; Zhu, L. Research on Data Augmentation for Object Detection Based on X-ray Security Inspection Picture. In Proceedings of the 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 25–27 August 2020.
26. Liu, S.; Chen, P.; Woźniak, M. Image enhancement-based detection with small infrared targets. *Remote Sens.* **2022**, *14*, 3232. [[CrossRef](#)]
27. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
28. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]
29. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer International Publishing: Berlin, Germany, 2014.
30. Qiao, S.; Chen, L.C.; Yuille, A. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. *arXiv* **2020**, arXiv:2006.02334.
31. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122.
32. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
33. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of Features. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
34. Guangdong Industrial Intelligent Manufacturing Innovation Competition [Competition 1]. Tianchi Competition—Ali Cloud Tianchi. Fabric Defect Detection Data Set. 2019. Available online: <https://tianchi.aliyun.com/dataset/79336> (accessed on 15 June 2022).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.