

## Research Article

# Detection of Microdefects in Fabric with Multifarious Patterns and Colors Using Deep Convolutional Neural Network

Rongfei Xia <sup>1</sup>, Yifei Chen,<sup>2</sup> and Yangfeng Ji<sup>1</sup>

<sup>1</sup>Chengyi College, Jimei University, 199 Jimei Avenue, Xiamen 361021, China

<sup>2</sup>School of Marine Engineering, Jimei University, 176 Shigu Road, Xiamen 361021, China

Correspondence should be addressed to Rongfei Xia; [rongfei.xia@jmu.edu.cn](mailto:rongfei.xia@jmu.edu.cn)

Received 5 October 2023; Revised 19 November 2023; Accepted 3 February 2024; Published 12 February 2024

Academic Editor: Nasir M. Ahmad

Copyright © 2024 Rongfei Xia 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.

Automatic detection of fabric defects is important in textile quality control, particularly in detecting fabrics with multifarious patterns and colors. This study proposes a fabric defect detection system for fabrics with complex patterns and colors. The proposed system comprises five convolutional layers designed to extract features from the original images effectively. In addition, three fully connected layers are designed to classify the fabric defects into four categories. Using this system, the detection accuracy is improved, and the depth of the model is shortened simultaneously. Optimal detection rates for testing dirty marks, clip marks, broken yams, and defect-free were 88.01%, 90.15%, 98.01%, and 97.73%, respectively. The experimental results show that the proposed method is effective, feasible, and has significant potential for fabric defect detection.

## 1. Introduction

Image classification is the foundation of defect classification using computer vision and plays a significant role in manufacturing product quality control. Defects on the fabric is one of the most important factors that affect the quality and price of textiles. Most defects on the fabric are caused by textile machine failures and problems in production. To the best of our knowledge, the detection of fabric defects remains primarily manual, which has low efficiency, high labor cost, and low accuracy. In addition, fabric defects are becoming increasingly difficult to detect because of the wide variety of fabrics and increasingly complex patterns and colors. With the advancement of science and technology, computer vision with higher detection accuracy and lower cost has gained widespread interest in fabric defect classification applications.

In recent years, several techniques have been proposed to improve the accuracy of fabric defect detection [1–4]. Kumar [5] presented a method for segmenting local textile defects based on a feed-forward neural network. This effective method offers a low-cost and single-PC-based solution for fabric defect classification. Pourkaramdel et al. [6] proposed a novel approach based on one dimensional local binary

patterns. The experimental results have showed that the proposed approach has a high detection rate for different surfaces such as stone, paper, and textile. Seçkin and Seçkin [7] proposed a new feature extraction method called intertwined frame vector feature extraction for defect detection. These features are used in machine learning classification algorithms. The experimental results have shown that the proposed method is faster and provides higher accuracy. Li and Zhang [8] proposed a novel automatic inspection scheme using smart visual sensors for warp knitting machines. The proposed scheme was effective, and its classification rate reached 98%. Çelik et al. [9] proposed a method adaptable to different fabric types based on a machine vision system. The system could detect defective areas in a denim sample with an average true classification rate of 91.7% and a false classification rate of 6.5%. These methods consist of two processes: First, they describe the image characters extracted from the original images and then use them as input to the classifier. This method typically relies on human feature selection, which significantly affects the classification results if the selected features are incomplete.

Many studies have demonstrated that deep learning performs excellently in feature learning and classification tasks.

Convolutional neural networks (CNNs) exhibit excellent performance in image recognition and classification [10, 11]. Usually, the accuracy of a CNN can be improved through various aspects, such as network architecture, nonlinear activation functions, supervision components, regularization mechanisms, and optimization techniques [12]. These improvements have led to further research on image classification using deep convolutional neural networks (DCNNs). DCNN has shown advantages and potential in various areas of classification tasks, such as traffic density measurement [13], medical diagnosis [14], visual recognition [15], and facial recognition [16, 17]. Recently, Jing et al. [18] proposed a modified AlexNet using CNN for the classification of yarn-dye fabric defects. The experimental results showed a promising performance with an acceptable average classification rate and strong robustness for yarn-dyed fabrics. Yao et al. [19] builds a deep learning defect detection network incorporating an attention mechanism. The number of samples of each defective type were enriched by the data augmentation strategy. The experimental results show that this algorithm can effectively detect 39 categories of fabric defects. Zhang et al. [20] proposed a lightweight network for defect detection in resource-constrained scenarios, the network improving small defect detection accuracy by applying the channel attention mechanism to extract more refined defect features under the complex background disturbance. Alruwais et al. [21] developed a novel hybrid mutation moth flame optimization with deep learning-based smart fabric defect detection technique for sustainable manufacturing. The experimental values demonstrate that the accuracy of the proposed method is 95.47%. However, it is difficult for these existing methods to achieve better adaptability, owing to the wide range of fabrics with complicated patterns and colors. The existing deep learning techniques are usually too complex to be run and trained on an industrial computer with limited computing resources.

This study aims to accurately detect microdefects in fabric with multifarious patterns and colors. To classify the defects, a model for feature extraction is proposed. These features are solely used for fabric defect classification without any other processing. The model uses batch normalization (BN) and inception to decrease the depth of the network with the same precision and enhance the training speed. The proposed model effectively detects microdefects in fabrics with various patterns and colors and can be used for automatic detection in fabric production lines.

## 2. Detection Algorithm for Fabric Defects

Inspired by the traditional AlexNet [22] and the application of other typical models [23, 24], the proposed model for fabric defect detection consists of BN and inception. BN has recently proven its effectiveness and importance in deep learning [25]. Each time a SGD is executed, the corresponding activation is normalized using a minibatch to make the mean of the computed result zero. Using  $x$  as the training sample, the BN formula is as follows:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E[x^k]}{\sqrt{\text{Var}[x^k]}}, \quad (1)$$

where  $k$  represents the  $k^{\text{th}}$  dimension of the data.  $E[x^k]$  and  $\sqrt{\text{Var}[x^k]}$  are the expectation and variance of training sample  $x$ , respectively.

The absolute value of  $\hat{x}^{(k)}$  is significantly small, and the overall data are located in the nonsaturation zone of the sigmoid. To solve this problem, an inverse transformation is performed to ensure the recovery of the original data using the following formula:

$$y^{(k)} = \gamma \hat{x}^{(k)} + \beta^k, \quad (2)$$

where  $\gamma$  and  $\beta$  are trainable parameters.

The accuracy of DCNN can be improved by adding nodes or layers. However, this leads to model overfitting and increases the computational cost. The inception model adds nodes to a DCNN while increasing the layers [26].

Figure 1 shows the architecture of the inception model. To eliminate the influence of the filter size on the recognition results, multiple features were extracted using filters of different sizes. The inception model uses  $1 \times 1$  convolutions,  $3 \times 3$  convolutions,  $5 \times 5$  convolutions, and max pooling of size  $3 \times 3$  convolutions in paper. Figure 1(a) shows a naive version of the inception model. However, the paper indicated that the parameters of the inception model filters are the sum of all branches, leading to a computational blowup within a few stages.

$1 \times 1$  convolutions can reduce the dimensionality of the output matrix and combine information in different channels. That is,  $1 \times 1$  convolutions can be used to compute reductions before the expensive  $3 \times 3$  convolutions,  $5 \times 5$  convolutions, and after max-pooling layers. Figure 1(b) shows the final inception model.

Based on our experience, the size of the input images is set to  $224 \times 224$  with three color channels. Next, the inception layer, which is stacked by  $1 \times 1$  convolution kernels,  $3 \times 3$  convolution kernels,  $5 \times 5$  convolution kernels, and max pooling with a filter size of  $3 \times 3$  follows, as shown in Figure 2. Then, average pooling with a filter size of  $3 \times 3$  is performed, which can reduce the estimator of variance caused by the limits of neighborhood size. Next, there are two fully connected layers of 256 neurons. A dropout follows the two fully connected layers to avoid overfitting. In the training procedures, the last fully connected layer has the same number of neurons as the number of output classes, which is the obtained output matrix. Subsequently, the defect type is then obtained using the SoftMax activation function. After the DCNN is trained, the SVM classifier replaces the SoftMax classifier and the last fully connected layer for the final classification. In this case, the size of the output feature map of the last fully connected layer is  $1 \times 1 \times 256$ .

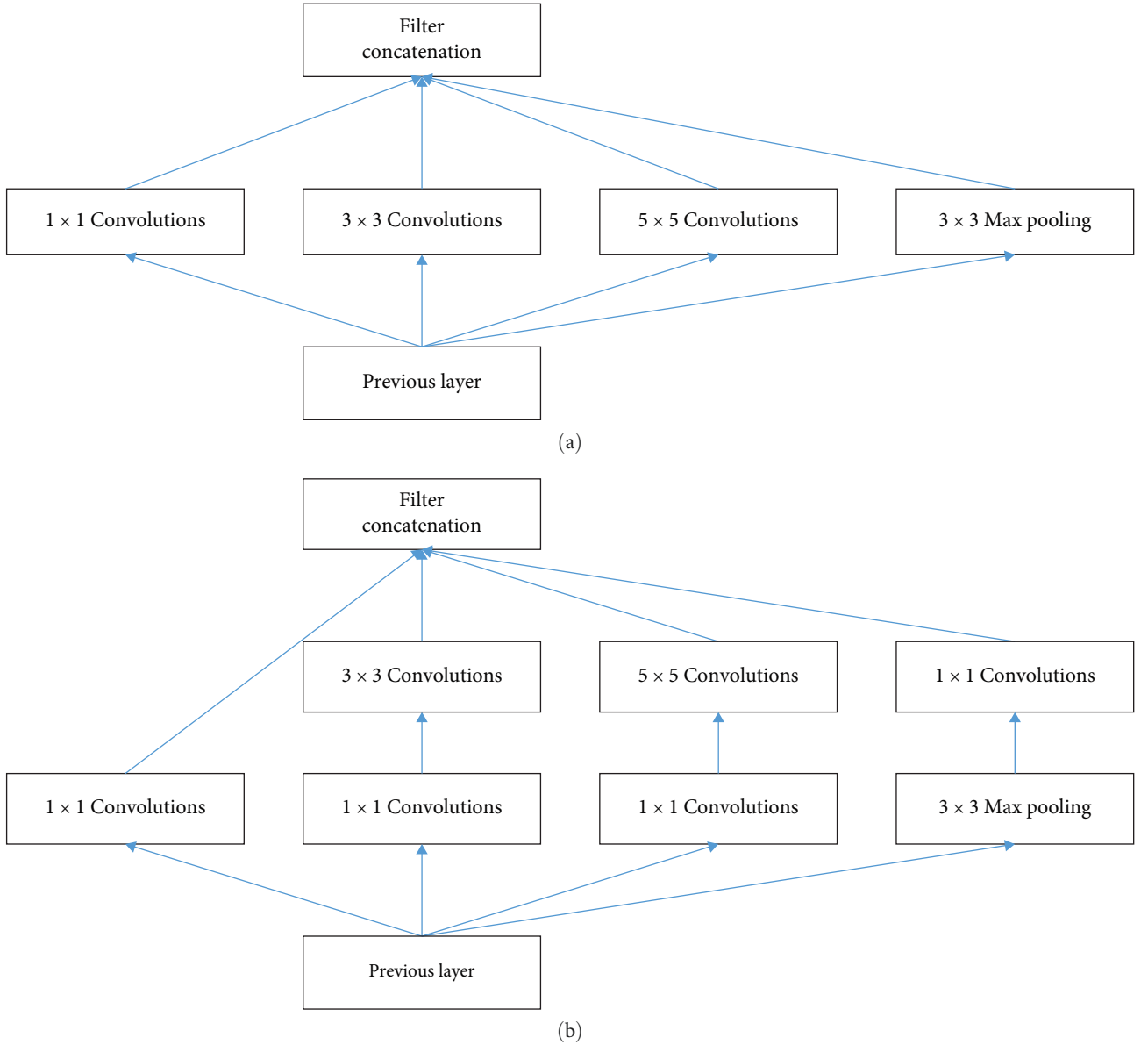


FIGURE 1: Inception module architecture. (a) Naive version of the inception model. (b) Inception module with dimensionality reduction.

The loss between the predicted and actual labels is calculated using sigmoid cross-entropy. The next step is to train the model and optimize the loss. Gradient descent is a typical practice for solving optimization problems. A gradient update is performed each time all database data are required. However, the update speed is slow, and the model cannot be updated online. Therefore, Bottou [27, 28] proposed a SGD, which can converge to the global minimum of the convex function and local minima of the nonconvex function. Using SGD, the update speed is fast, and the model can be updated online. Despite the extensive use of gradient-based optimization techniques for DCNN and image classification, the method remains a challenge because of its substantial limitations. Therefore, improved optimization techniques have been proposed, including Momentum [29], Adagrad [30],

Adadelta [31], RMSprop [32], and adaptive moment estimation (Adam) [33]. The advantage of Adam is that the learning rate is restricted to a limited scope in each iteration by the bias correction method to enable comparatively stable parameters. Hence, the loss function is  $f(\theta)$  and its gradient is  $g_t = \nabla \theta_{t-1} f(\theta_{t-1})$ .  $\theta$  is a parameter. Based on the concept of momentum in physics, the gradient is replaced by accumulated momentum, resulting in the following formula:

$$m_t = \beta_1 \times m_{t-1} + (1 - \beta_1) \times g_t, \quad (3)$$

$$n_t = \beta_2 \times n_{t-1} + (1 - \beta_2) \times g_t^2, \quad (4)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad (5)$$

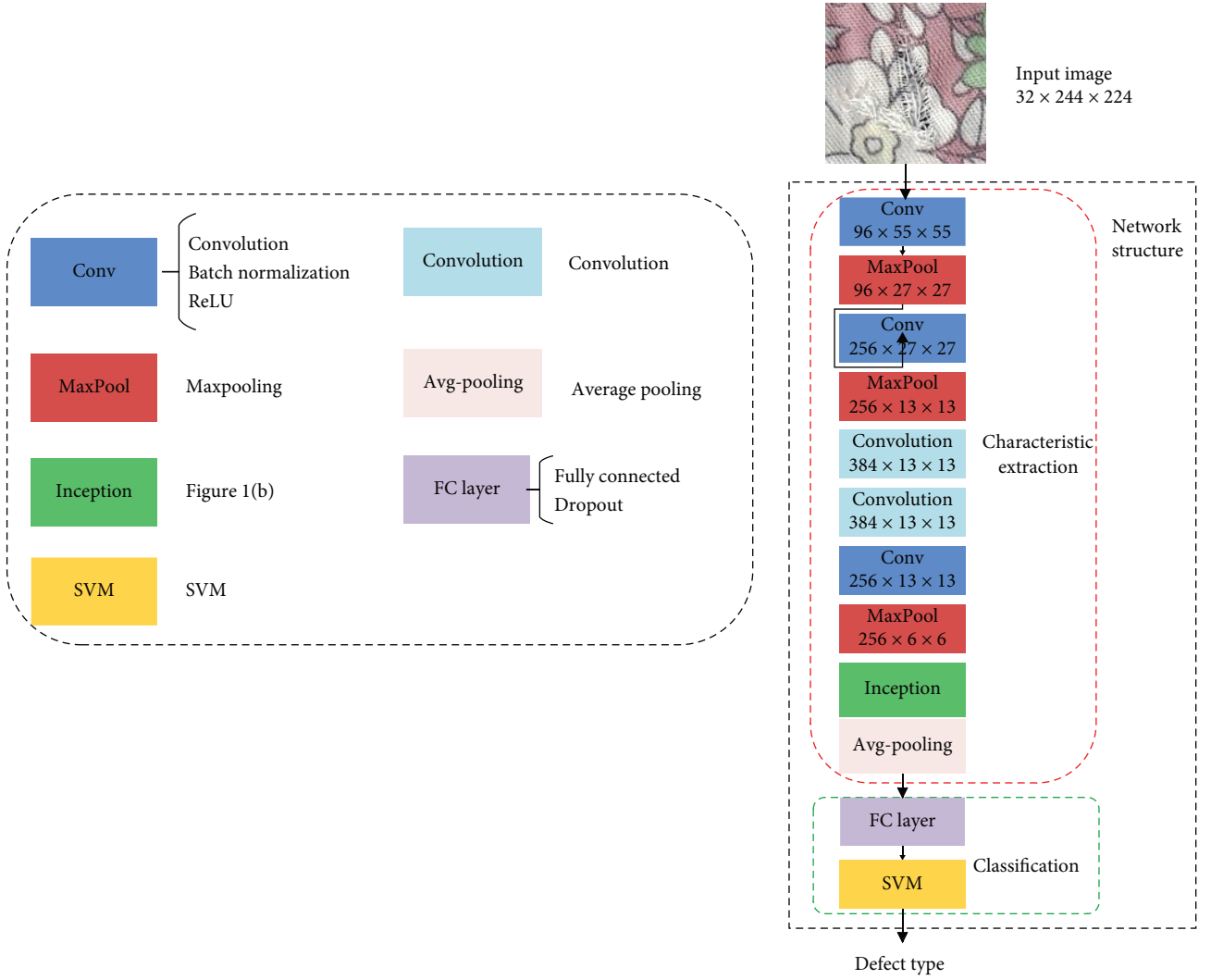


FIGURE 2: Architecture of proposed DCNN.

$$\hat{n}_t = \frac{n_t}{1 - \beta_2^t}. \quad (6)$$

$m_t$  and  $n_t$  denote the mean and variance of the gradient, respectively.  $\hat{m}_t$  and  $\hat{n}_t$  modify  $m_t$  and  $n_t$ , respectively. When  $\beta_1$  is 0.9 and  $\beta_2$  is 0.9999, the parameter update formula is as follows:

$$\theta_{t-1} = \theta_t - \frac{\hat{m}_t}{\sqrt{\hat{n}_t + \epsilon}}. \quad (7)$$

It was found that the gradient estimate does not require additional memory and is dynamically adjusted by the gradient. In addition, the learning rate is dynamically constrained by  $-\frac{\hat{m}_t}{\sqrt{\hat{n}_t + \epsilon}}$ .  $\epsilon$  is used to ensure that the denominator is a nonzero value.

### 3. Experiment

**3.1. Image Dataset and Evaluation Criteria.** Samples with fabric defects were obtained from an umbrella factory.

Generally, there are four types of fabric defects: dirty marks, clip marks, broken yams, and defect-free defects. A fabric dataset, including large and small datasets, was created. The large dataset contained 4,000 fabric images, including four defect types: dirty marks, clip marks, broken yams, and defect-free. The images were randomly divided into training and validation sets in a 5 : 1 ratio. A small dataset containing 2,000 fabric images was selected for the testing sets, and it included four classes: dirty marks, clip marks, broken yams, and defect-free. Typical defect samples in the fabric datasets are shown in Figure 3.

The improved defect inspection equipment is shown in Figure 4. The fabric images were captured using a line-scan digital camera. Furthermore, the line light source is behind the fabric. Image storage and inspection are performed by a computer.

To evaluate the performance of the fabric defect detection system, accuracy (ACC), false positive rate (FPR), and recall rate are adopted. ACC, FPR, and recall rate are defined as Equations (8), (9), and (10), respectively. A high FPR repeatedly stops the machine because of a false alarm,



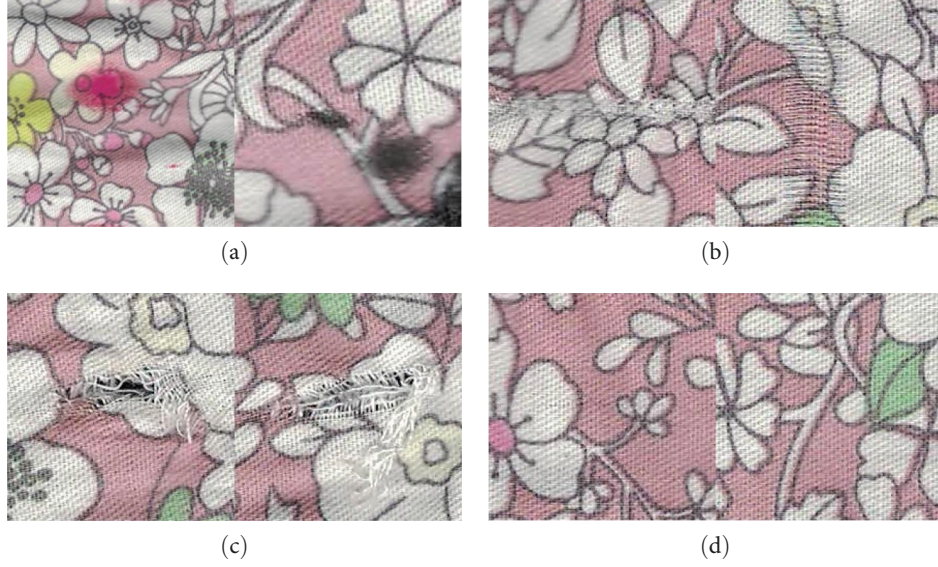


FIGURE 3: Fabric defect types: (a) dirty marks; (b) broken yams; (c) clip mark; and (d) defect-free.



FIGURE 4: Defect inspection equipment.

significantly affecting the generation efficiency. A low recall rate can result in defective products going to market. Furthermore, a low recall rate affects product quality.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \quad (8)$$

$$FPR = \frac{FP}{FP + TN} \times 100\%, \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \times 100\%. \quad (10)$$

where TP is the number of defective samples classified correctly, TN is the number of nondefective samples classified correctly, and FN and FP are the numbers of defective and nondefective samples falsely detected, respectively.

**3.2. Training and Testing.** The training set was used to train the proposed DCNN model. The choice of training parameters was as follows: (1) An optimization algorithm was used

TABLE 1: Performance of proposed DCNN for 2,000 iterations.

Learning rate	ACC (%)
0.01	86.9
0.001	92.7
0.0001	87.3

to optimize the parameters of the model, and four gradient descent variants were used as optimizers to compare the performance of the proposed method. (2) The learning rate controls the training speed of the network weights and must be set to a value within a range to ensure that the model has good performance. Learning rates of 0.01, 0.001, and 0.0001 were used to train the model. (3) A batch size of 32 was set based on the sample size of fabric defects. (4) The epoch is a cycle period that includes one forward and one backward transmission for training all samples. The following formula was used:  $\text{epoch} = (\text{Total sample}/\text{batch size})/\text{iteration} = 1$ . We expected to achieve the required results by training the model through multiple iterations.

## 4. Results and Discussion

The performance of the proposed DCNN model was evaluated using the training, validation, and testing sets. The input samples of fabric defects were resized to  $224 \times 224$ , and the evaluation was conducted with learning rates of 0.01, 0.001, and 0.0001; the number of iterations was set to 2,000. Table 1 shows the recognition performance using different learning rates. The learning rate of 0.001 yielded the highest performance.

To demonstrate the effectiveness of the proposed DCNN model, BN and inception were removed from the model. The learning rate of the models is 0.001. The performances are presented in Table 2. As shown in Figure 5, removing either BN or inception did not yield better accuracy.

TABLE 2: Comparison of the performance of models without BN or inception.

Model	ACC (%)	FPR (%)	Recall (%)
Removed BN	87.5	1.3	90.5
Removed inception	52.3	5.2	68.2
Proposed DCNN model	92.7	0.6	99.7

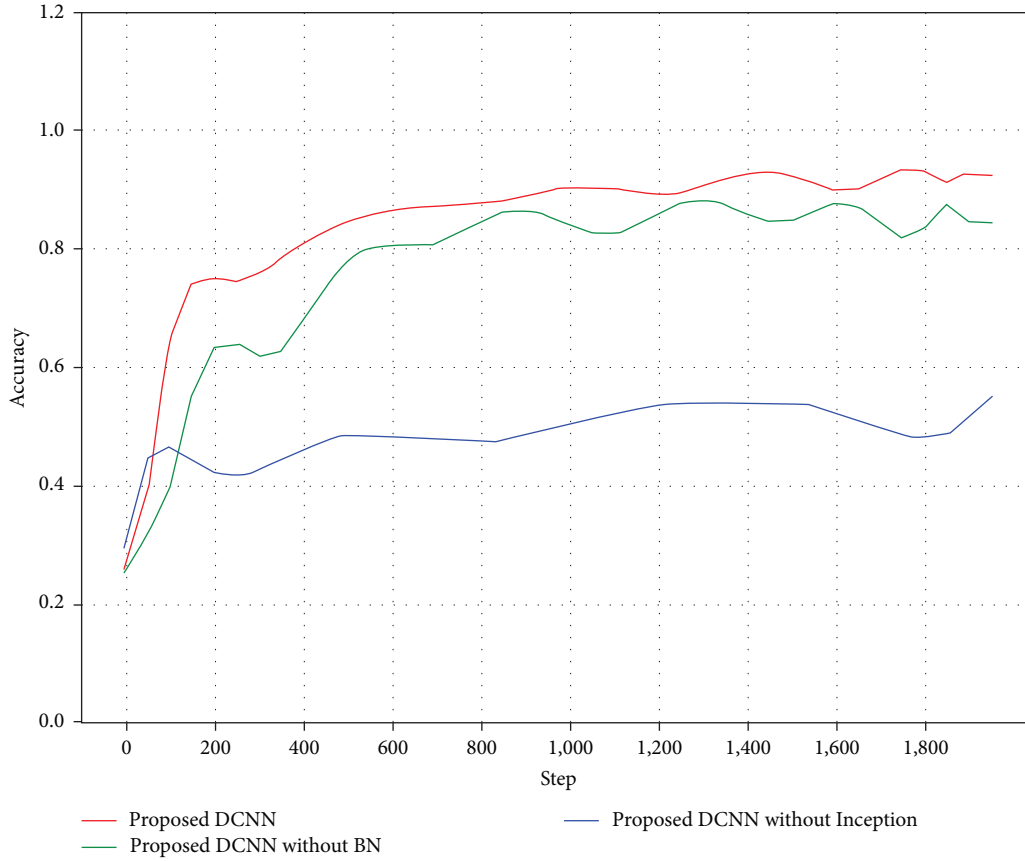


FIGURE 5: Performance of models without BN or inception.

TABLE 3: Classification accuracy using different optimizers.

Optimizer	ACC (%)	FPR (%)	Recall (%)
SGD	62.7	2.3	85.6
Momentum	84.5	1.3	86.2
RMSProp	90.4	1.2	99.2
Adam	92.7	0.6	99.7

Table 3 lists the detection and false detection rates of the proposed model with different optimizers. The detection accuracy of the proposed model using different optimizers is shown in Figure 6. When SGD and Momentum were used as optimizers, the proposed DCNN model exhibited low classification accuracy. When Adam and RMSProp were used as optimizers, the detection accuracy and recall rate of the proposed model were high and similar; however, RMSProp exhibited a higher FPR.

In addition, AlexNet is a traditional DCNN with good performance in classification tasks, whereas YOLOv8 is a state-of-the-art image recognition method. Therefore, the performance of the proposed model was compared with AlexNet and YOLOv8 and evaluated. The hyperparameters of the models are listed in Table 4, and the results are listed in Table 5. Our proposed model exhibited a lower accuracy and recall rate than that of YOLOv8; however, in practical applications, real-time detection is a key index, which directly

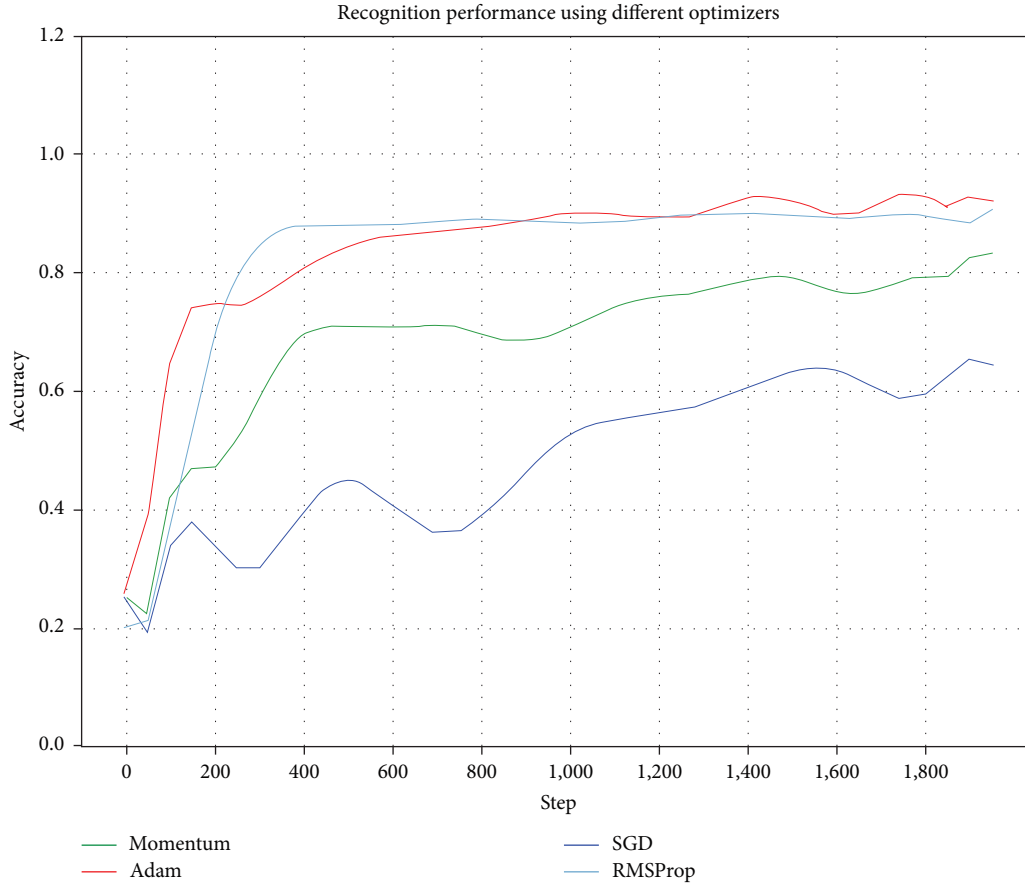


FIGURE 6: Detection performance using different optimizers.

TABLE 4: Hyperparameters of the models.

Parameters	AlexNet	YOLOv8	Proposed DCNN model
Training option	Adam	SGD	Adam
Batch size	135	64	32
Learning rate	0.001	0.01	0.001
Epoch	15	160	20
Weight decay	—	0.0005	—

TABLE 5: Performance of the proposed model compared with AlexNet.

Model	ACC (%)	FPR (%)	Recall (%)	Runtime (ms)
AlexNet	80.9	6.6	87.5	120
YOLOv8	95.3	0.3	99.7	580
Proposed DCNN model	92.7	0.6	99.7	135

affects production efficiency. Therefore, the network structure proposed in this paper is relatively simple, so the running time of the proposed method can meet the real-time requirements of application scenarios. The experiments in this work are based on the GeForce GTX 1080Ti, and the runtime is 135 ms, which meets the real-time demand.

In the above experiments, the proposed DCNN model exhibited excellent performance using a batch size of 32 and

a learning rate of 0.001. To generalize our experiment and results, we repeated our experiments 10 times using the validation and testing sets and calculated the average of all times as the final classification accuracy. Table 6 presents the results.

We calculated the number of false detections for different types of fabric defects in the testing set, as shown in Figure 7, and the false detection rates for different types of fabric defects are listed in Table 7.

TABLE 6: Detection rates on validation samples and test samples.

Run ID		1	2	3	4	5	6	7	8	9	10	ACC
Validation accuracy (%)		91.66	92.11	92.84	93.15	92.58	92.70	92.75	93.23	92.68	92.93	92.66
Test accuracy (%)	Dirty marks	86.91	87.22	86.73	87.12	87.54	88.01	86.45	87.26	87.51	86.73	87.15
	Clip mark	86.52	88.71	87.38	85.82	89.07	89.15	89.46	90.05	85.75	90.15	88.21
	Broken yam	97.35	98.44	98.12	98.32	97.96	98.02	98.73	98.16	98.01	97.79	98.09
	Defect-free	96.44	97.58	97.36	97.48	96.62	96.49	96.94	97.70	97.01	97.73	97.14

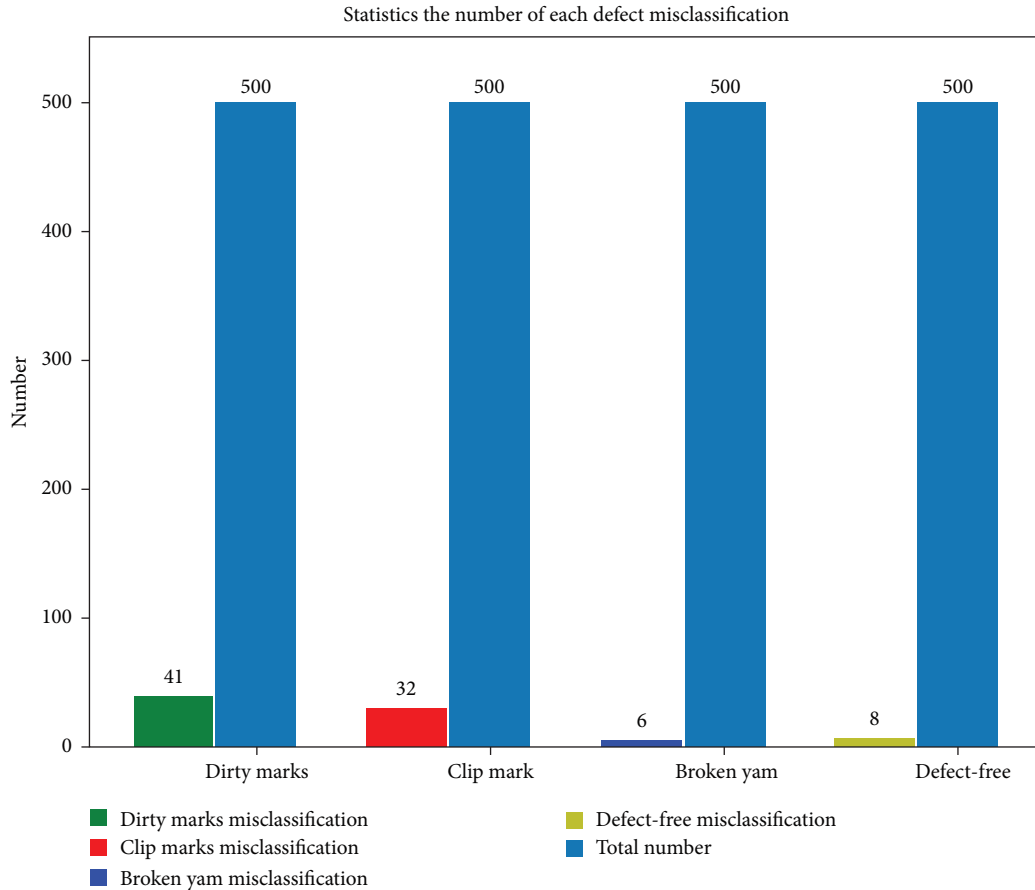


FIGURE 7: Number of each defect misclassification.

TABLE 7: Misclassification rate for each fabric defect.

Fabric defect	ACC (%)
Dirty marks	91.8
Clip mark	93.6
Broken yam	98.9
Defect-free	98.4

In the experiments, better results were achieved in terms of the detection rates of the datasets. This proves that the proposed DCNN model performs well regarding detection rate. However, a low detection rate was still obtained for dirty and clip marks, which is possible because the characteristics of dirty and clip marks are similar.

## 5. Conclusion

Deep learning is an effective image-classification method. This study proposed a fabric defect detection model based on a DCNN to ensure the quality of fabrics with multifarious patterns and colors. The application to fabrics with multifarious patterns and colors showed that the proposed DCNN model achieves good accuracy for fabric defect detection. The experimental results showed that the proposed model can achieve a detection accuracy of 92.7%.

Although the proposed DCNN model exhibited good performance, it still requires further improvement that can be covered in future studies. (1) The number of samples is not sufficient; hence, a database of other defects, such as hanging threads and buttonhole selvage, is required. (2)



Increase the speed of the training process and improve accuracy by optimizing the network. (3) The network is optimized using optimization techniques such as initialization schemes and skip connections.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Young and Middle-Aged Teacher Education Research Program of Fujian Province (grant no. JAT201037) and Doctoral Research Fund Program of Chengyi College, Jimei University (grant no. CK21018).

## References

- [1] Z. Liu, B. Wang, C. Li, M. Yu, and S. Ding, "Fabric defect detection based on deep-feature and low-rank decomposition," *Journal of Engineered Fibers and Fabrics*, vol. 15, no. 2, pp. 1–12, 2020.
- [2] H. Uzen, M. Turkoglu, and D. Hanbay, "Texture defect classification with multiple pooling and filter ensemble based on deep neural network," *Expert Systems with Applications*, vol. 175, Article ID 114838, 2021.
- [3] J.-F. Jing, H. Ma, and H.-H. Zhang, "Automatic fabric defect detection using a deep convolutional neural network," *Coloration Technology*, vol. 135, no. 3, pp. 213–223, 2019.
- [4] S. Fekri-Ershad and F. Tajeripour, "A robust approach for surface defect detection based on one dimensional local binary patterns," *Indian Journal of Science and Technology*, vol. 5, pp. 3197–3203, 2012.
- [5] A. Kumar, "Neural network based detection of local textile defects," *Pattern Recognition*, vol. 36, no. 7, pp. 1645–1659, 2003.
- [6] Z. Pourkaramdel, S. Fekri-Ershad, and L. Nanni, "Fabric defect detection based on completed local quartet patterns and majority decision algorithm," *Expert Systems with Applications*, vol. 198, Article ID 116827, 2022.
- [7] A. C. Seçkin and M. Seçkin, "Detection of fabric defects with intertwined frame vector feature extraction," *Alexandria Engineering Journal*, vol. 61, no. 4, pp. 2887–2898, 2022.
- [8] Y. Li and C. Zhang, "Automated vision system for fabric defect inspection using Gabor filters and PCNN," *SpringerPlus*, vol. 5, Article ID 765, 2016.
- [9] H. Çelik, M. Topalbekitoglu, and L. Dülger, "Real-time denim fabric inspection using image analysis," *Fibres and Textiles in Eastern Europe*, vol. 23, no. 3, pp. 85–90, 2015.
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [11] Y. Chen, R. Xia, and Y. Feng, "The research of chronic gastritis diagnosis with electronic noses," *Journal of Sensors*, vol. 2021, Article ID 5592614, 8 pages, 2021.
- [12] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [13] J. Chung and K. Sohn, "Image-based learning to measure traffic density using a deep convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1670–1675, 2018.
- [14] U. Raghavendra, H. Fujita, S. V. Bhandary, A. Gudigar, J. H. Tan, and U. R. Acharya, "Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images," *Information Sciences*, vol. 441, pp. 41–49, 2018.
- [15] K. Seeliger, M. Fritsche, U. Güçlü et al., "Convolutional neural network-based encoding and decoding of visual object recognition in space and time," *NeuroImage*, vol. 180, pp. 253–266, 2018.
- [16] S. Ravidas and M. A. Ansari, "Analysis of multi-view face detection using deep convolutional neural networks," *Indian Journal of Industrial and Applied Mathematics*, vol. 9, no. 2, Article ID 231, 2018.
- [17] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller, "One-to-many face recognition with bilinear CNNs," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9, IEEE, Lake Placid, NY, USA, 2016.
- [18] J. Jing, A. Dong, P. Li, and K. Zhang, "Yarn-dyed fabric defect classification based on convolutional neural networks," *Optical Engineering*, vol. 56, no. 9, Article ID 093104, 2017.
- [19] H. Yao, Q. Na, S. Zhu, M. Lin, and J. Yu, "Fabric defect detection with an attention mechanism based on hard sample training," *Textile Research Journal*, vol. 92, no. 9–10, pp. 1429–1443, 2022.
- [20] J. Zhang, J. Jing, P. Lu, and S. Song, "Improved MobileNetV2-SSDLite for automatic fabric defect detection system based on cloud-edge computing," *Measurement*, vol. 201, Article ID 111665, 2022.
- [21] N. Alruwais, E. Alabdulkreem, K. Mahmood et al., "Hybrid mutation moth flame optimization with deep learning-based smart fabric defect detection," *Computers and Electrical Engineering*, vol. 108, Article ID 108706, 2023.
- [22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [23] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, vol. 2015, Article ID 258619, 12 pages, 2015.
- [24] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, no. 6, pp. 378–384, 2017.
- [25] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, ACM, 2015.
- [26] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, IEEE, Boston, MA, USA, 2015.
- [27] L. Bottou, "Online learning and stochastic approximations," in *On-Line Learning in Neural Networks*, pp. 9–42, Cambridge University Press, 1998.
- [28] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Physica-Verlag HD, 2010.
- [29] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *Proceedings of Machine Learning Research*, vol. 28, no. 3, pp. 1139–1147, 2013.
- [30] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

- [31] M. D. Zeiler, “ADADELTA: an adaptive learning rate method,” Computer Science, 2012.
- [32] T. Tieleman and G. Hinton, “Lecture 6.5—RMSProp, COURSERA: neural networks for machine learning,” Technical Report, 2012.
- [33] D. Kingma and J. Ba, “Adam: a method for stochastic optimization,” Computer Science, 2014.