Fabric defect detection method based on Cascade Deep Support Vector Data Description

Xupeng Wang¹, Yueyang Li*¹, Haichi Luo²

¹Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, China
²College of Internet of Things Engineering, Jiangnan University, Wuxi, China

*CONTACT: Yueyang Li, lyueyang@jiangnan.edu.cn

Abstract. The Fabric defect detection method based on Cascade Deep Support Vector Data Description (SVDD) is proposed in this paper. The method describes the data by Deep SVDD to realize the correct evaluation between the normal fabric images and the images with defects in the high dimensional space. Combining with the cascading method, the proposed algorithm can be used to detect minor defects more accurately and quickly. We take the anomaly score as the evaluation result of the testing image, then the Median Absolute Deviation (MAD) outlier detection method is applied to get the final defect detection results. This newly developed method can be trained with only a small amount of defect-free samples, which greatly reduces manual intervention. A variety types of defects in the general fabric images can be detected efficiently. The experimental results demonstrate that the proposed model has good global performance and high recall rate.

1. Introduction

Defect detection is a key step in textile industry quality inspection. At present, fabric detection is progressing from manual method to machine vision method. During the processing of the traditional machine vision methods, the feature of fabric defects can be extracted by many operators. However, we have to design different operators in order to efficiently extract effective features from fabric images with various types of defects according to experts’ experience.

In recent years, the fabric defect detection method based on deep learning shows strong performance and the features learned by Convolutional Neural Network (CNN) can accurately complete the target recognition task. Compared with the traditional manual feature extraction, these features have higher universality and portability. Masci et al. compared Max-pooling CNNs with SVM algorithm in the detection of seven defects in cold strips, showing the outstanding ability of feature extraction of CNNs [1]. Liu Rao et al. combined migration learning and CNN model to fuse features extracted by AlexNet and GoogLeNet, and finally added Support Vector machine (SVM) classifier for defect classification [2]. These algorithm needs enough defect fabric images to build a model. However, in practical applications, it is costly to obtain various kinds of fabric images with defects. In view of this problem, Wang et al. [3] proposed a detection method combining gray-level gradient co-occurrence matrix (GGCM) with Support Vector Data Description (SVDD) [4]. Only normal fabric images are used during the training. However, this is a hybrid approach to feature extraction and classification tasks, the complex model is time consuming and it cannot realize fast end-to-end detection. Ruff et al. proposed an anomaly detection method Deep Support Vector Data Description,
which is trained on an anomaly detection based objective[5]. The common variation factor is extracted by the network to minimize the volume of the hypersphere. The network maps the data points closely to the center of the sphere. Deep SVDD is an end-to-end detection method. However, it is not effective to directly use this method for fabric defects detection, especially for detecting the small defects.

This paper presents a new learning method for fabric defect detection based on Cascade Deep SVDD. The main contributions of this paper are summarized as follows. Firstly, we propose a new method based on Cascade Deep SVDD for fabric defect detection, compared with the traditional method to extract features manually, the deep convolutional network can enable the detection model to learn more rich texture features. Secondly, there is no need to find a large number of fabric images with defects. Only a small number of normal samples are needed to complete the training. Thirdly, the method has good universality, which has good detection effect for most kinds of fabric defects. Finally, using cascade strategy can improve the detecting accuracy for small defects and it can also increase the detection speed. The experimental results demonstrate its good detecting performance.

2. Methods

We provide a new fabric defect detection method based on Cascade Deep SVDD. Firstly, through slice the original flawless image, we make two batches of patches with different dimensions as training data. The process is described in section 2.4.1 in detail. The training data are totally derived from flawless fabric sample images. The large patches are used for training the model1 and small patches for the model2. In each model, by training the parameters $W$ of the convolutional neural network, the image patches can be mapped into high-dimensional space and a minimum volume hypersphere model can be built. During the testing, the anomaly scores, which evaluate the distance between the testing patches and the center of the hypersphere, can be obtained. Then, by using the MAD method, we can get the defective patches. All the large patches could be checked by the Model1 and only the patches with large anomaly score value will be detected by the Model2. Finally, we can use the same method to obtain the defective small patches by Model2. The overall architecture of the Cascade Deep SVDD network model is shown in Figure 1.

![Figure 1. Overall architecture of the proposed Cascade Deep SVDD model.](image)

2.1. SVDD

SVDD is an one-class classification algorithm. The objective of SVDD is to find the smallest hypersphere with center $c \in F_k$ and radius $R > 0$ that encloses the majority of the data in feature space $F_k$. The SVDD primal problem is given by:

$$
\min_{R,W} \; R^2 + \frac{1}{Vn} \sum \xi_i
$$

s.t. 
$$
\|\phi_i(x_i) - c\|_{\nu_i}^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \; \forall i.
$$
slack variables $\xi_i \geq 0$ allow a soft boundary and hyperparameter $\nu \in (0,1]$ controls the trade-off between penalties $\xi_i$ and the volume of the sphere. Points which fall outside the sphere, i.e. $\|\phi_i(x_i) - c\|_2 > R^2$, are deemed anomalous.

2.2. Deep SVDD

Based on deep learning, Deep SVDD is a fully deep approach for Anomaly Detection (AD). Deep SVDD learns to extract the common factors of variation of the data distribution by training a neural network to fit the network outputs into a hypersphere of minimum volume. Unlike other methods of Deep AD, Deep SVDD does not rely on reconstruction errors or use a hybrid method to complete detection, but can be directly used to express learning and detection from end to end. The principle diagram of SVDD method is shown in Figure 2.

![Figure 2. The method schematic diagram of Deep-SVDD.](image)

Different from the kernel-based SVDD method, Deep SVDD seeks for the minimum capacity estimation through the neural network by finding a data-enclosing hypersphere with the minimum size. With Deep SVDD, we learn useful feature representations of the data together with the one-class classification objective. To do this we employ a neural network that is jointly trained to map the data into a hypersphere of minimum volume. Deep SVDD has hard boundary and soft boundary modes. Hard boundaries map all outliers outside the hypersphere, while the soft boundary model allows some crossing points. For fabric defect detection, the hard boundary model can be applied to get good results. Hard boundary Deep SVDD algorithm maps the sample points in space A into the hypersphere in space B by learning network parameters $W$. The optimization objective is as follows:

$$\min_{W} \frac{1}{n} \sum_{i=1}^{n} \|\phi_i(x_i;W) - c\|_2^2 + \frac{\lambda}{2} \sum_{l=1}^{L} \|W_l\|_F^2$$

(2)

Where $n$ is the number of sample sets, $x_i, x_2, ..., x_n$ is the sample data, $W$ is the network learning parameter, $c$ is the center of the hypersphere. Function $\phi_i(x_i;W)$ is the characteristic representation of neural network parameters. The first item employs a quadratic loss for penalizing the distance of every network representation. The second term is the regularization term. The parameter $\lambda(\lambda > 0)$ controls the weight of regularization and $L$ is the total number of layers of the network, $l \in \{1, ..., L\}$. $W_l$ is the weight parameter of each layer, where $\| \cdot \|_F$ denotes the Frobenius norm. Optimization formula (1) obtains network parameters $W^*$ and $R^*$ makes the data points map tightly to the central range of the hypersphere.

2.3. Anomaly score

For a given test point $x \in B$, we define an anomaly score $S(x)$ based on the distance of the point to the center of the hypersphere, that is:
Where $W^*$ is the trained model network parameter. In hard-bounded Deep SVDD model, you can get an anomaly score $S(x)$. In this way, all the points outside the hypersphere will get a high anomaly score, while all the points inside the hypersphere will get a low anomaly score.

In the method of Deep SVDD, the network parameters $W^*$ completely characterize a Deep SVDD model and no data must be stored for prediction, thus endowing Deep SVDD a very low memory complexity.

2.4. Cascading Deep SVDD

In this section, we introduce a cascade detection method, which can detect fabric image with different resolutions. Considering the defect has only a small area in fabric image, we first used large image patches to filter out most of the original image, and then we used a small image patches to locate the defect area. This cascading approach provides faster detection speed and better accuracy.

2.4.1. Patch Extraction. Firstly, we need to obtain image patches for training. Here, an fabric image with the size of 800×600 pixels is taken as an example. For training the model1, we can get 4×3 image patches with the size of 200×200 pixels shown in Figure 3. For training the model2, 5×5 image patches can be obtained with the size of 40×40 pixels shown in Figure 4.

![Image 3](image3.png)  ![Image 4](image4.png)

Figure 3. The large image patches for training in the model1.  Figure 4. The small image patches for training in the model2.

2.4.2. Image Preprocessing. Each image patch is resized to the 32×32 pixels to satisfy for CNN network input. We pre-process all image patches with global contrast normalization and they are rescaled to [0, 1] via min-max-scaling. The purpose of normalization is to realize data centralization and improve the generalization ability of the model.

2.4.3. Training. Considering the trade-off between speed and accuracy, we use the CNN network based on the improved LeNet [6] as the training network. The final full link layer of the improved LeNet network is removed. Through experiments, we set the dimension of the eigenvectors of the last layer to 1024 dimensions. The LeNet network model with the shallow layer can be used to achieve a better detection effect, which greatly reduces the computation.

During the training, the network parameter $W$ are optimized using the improved Adam method[7]. The hyper parameters are shown in Table 1.

|                | The model1 training | The model2 training |
|----------------|---------------------|---------------------|
| initial learning rate | 1e-4               | initial learning rate | 1e-5               |
| optimizer       | Adam                | optimizer            | Adam                |
epochs | 200 | epochs | 500
---|---|---|---
batch size | 128 | batch size | 64
regularized weight | 0.1 | regularized weight | 0.1
alpha of Leaky Relu | 0.1 | alpha of Leaky Relu | 0.1

2.4.4. Testing and threshold determination. During the testing, the method of sliding window was applied. In the model1 training, 4×3 large patches are generated. The anomaly score of each patch could be obtained. If a large patch is detected to have defects, 5×5 small patches with size of 40×40 pixels can be obtained on this large defect patch. Each small patch has a corresponding anomaly score. Finally, MAD was applied to determine the segmentation threshold to complete the defect detection of the entire image [8]. The threshold is an important parameter to determine whether an image patch has defects or not.

In statistics, MAD is a robust measure of the sample deviation of univariate numerical data. The method is to determine whether each element is an outlier by finding whether the deviation between an element and the median value is within a reasonable range. The threshold $T$ is calculated as follows:

$$
T = \begin{cases} 
X_m + nM & \text{if } X_i > X_m + nM \\
X_m - nM & \text{if } X_i < X_m - nM 
\end{cases}
$$

(4)

The parameter $X_m$ is the median of all values. $bias = |X_i - X_m|$ is the absolute deviation between all elements and the median. The parameter $M$ is the median value of the absolute deviation, $MAD = bias_m$. The parameter $n$ is set to 3 in our experiment.

2.4.5. Experiment results. In order to make the results more intuitive, we add a yellow mask ([255,255,0] for RGB channel) on the original image. We normalize the anomaly score to [0,1]. The transparency of a completely abnormal image mask is 1, and that of a completely normal image mask is 0. The model1 test results are shown in Figure 5. The mask of the anomaly score is shown in Figure 5(b). The abnormal image patches can be clearly seen from the mixed image in Figure 5(c). We can use the same method for testing in the model2 test with smaller image patches. The model2 test results are shown in Figure 6. The final defect detection result of the original image is shown in Figure 6(c).
3. Experiments and Discussion

3.1. Datasets
In order to verify the defect detection performance of our method, the test images with the size of 800×600 pixels are all from a production line. Some samples are shown in Figure 7.

![Fabric Image Samples](image)

Figure 7. Experimental fabric image samples.

3.2. Evaluation Criteria
The general criteria for determining whether a fabric image is defective or not include precision rate $P$, recall rate $R$ and detection accuracy $Acc$. These criteria are defined as follows:

$$
P = \frac{TP}{TP + FP}$$

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

$$
Acc = \frac{(TP + TN)}{(TP + FN + TN + FP)}$$

(5)

Where $TP$ represents the correct number of inspected defective patches, while $FN$ represents the number of incorrectly detected defective patches. $TN$ represents the number of correctly detected flawless patches, and $FP$ represents the number of incorrectly detected flawless patches. Moreover, we also provide the running time of the proposed method for meeting the industrial requirements.

3.3. Results and Discussion
In Figure 8, (a)-(j) are the original images and test results for various common defects.

![Defect Images](image)
Figure 8. Experimental fabric image samples.

Figure 9 shows the experiments results for six fabric images with various kinds of defects. For example, Figure 9(a) shows the original Blue image, anomaly score mask image and the test result image, respectively. In the resulting image, there are 9 small image patches were determined to be defective and the rest area is defect-free. The details of experiment results are shown in Table 2. As to the Blue image, the test results shown in the second row indicate that the total number of the small image patches for testing is 540 and 88 small image patches are defective. The accuracy rate is 0.920 and a recall rate is 0.934.

Figure 9. Test results for some fabrics.

Table 2. Test results of various fabric pieces

| Fabric Color      | Number of defective image patches (All image patches) | Precision | Recall | Average detection speed(sheet/s) |
|-------------------|-------------------------------------------------------|-----------|--------|----------------------------------|
| Blue white        | 184(840)                                              | 0.942     | 0.921  | 0.34                             |
| Blue              | 88(540)                                               | 0.920     | 0.934  | 0.35                             |
| Dark blue         | 523(10100)                                            | 0.893     | 0.961  | 0.33                             |
| Lavender          | 239(1070)                                             | 0.912     | 0.924  | 0.29                             |
| Light gray        | 793(10900)                                            | 0.897     | 0.934  | 0.35                             |
| Origin            | 278(6000)                                             | 0.896     | 0.905  | 0.25                             |
| White blue        | 319(10800)                                            | 0.926     | 0.895  | 0.28                             |
| White             | 462(13500)                                            | 0.899     | 0.953  | 0.33                             |
| Average           |                                                        | 0.911     | 0.928  | 0.315                            |
4. Conclusions
In this paper, a new fabric defect detection method based on Cascade Deep SVDD was proposed. The experimental results show that we only use a small number of positive samples during the training. During the testing, various types of defects can be identified with a certain accuracy.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (51405198). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal used for this research.

References
[1] Masci J, Meier U, Ciresan D, et al. Steel defect classification with max-pooling convolutional neural networks[C]. (2012) Neural Networks (IJCNN), The 2012 International Joint Conference on. IEEE: 1-6.
[2] Liu Rao. (2019) Application of convolutional neural network in Textile Defect Detection [D]. Xi'an University of Technology, Xi'an.
[3] Wang Mengtao, LI Yueyang, DU Shuai, JIANG Gaoming, LUO Haichi. (2012) Fabric defect detection based on Gray gradient symbiosis matrix and SVDD [J]. Silk,55(12):50-56.
[4] Tax, D. M. J. and Duin, R. P. W. Support Vector Data Description. (2004) Machine learning, 54(1):45–66.
[5] Ruff, L., Vandermeulen, R. A., Nico Görnitz, Deecke, L., & Kloft, M. (2018). Deep One-Class Classification. International Conference on Machine Learning.
[6] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X. and Wang, G. (2015). Recent advances in convolutional neural networks, arXiv preprint arXiv:1512.07108.
[7] Kingma, D. P., & Ba, J. (2014). Adam: a method for stochastic optimization. Computer Science.
[8] Everitt, Brian S. (2008). Encyclopedia of statistics in behavioral science /. John Wiley & Sons.