



Automatic fabric defect detection in textile images using a labview based multiclass classification approach

T. Meeradevi¹ · S. Sasikala¹

Received: 24 January 2023 / Revised: 30 November 2023 / Accepted: 29 December 2023 /
Published online: 19 January 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Nowadays the detection of fabric defects is an active research topic to detect and resolve the difficulties faced in processing fabric in printing and knitting in textile industries. The traditional approach of visual screening of human fabrics is exceedingly time consuming and it is not reliable as it is much susceptible to human errors. There are two major issues in defect inspection like defect identification and classification in fabric. Automatic identification of defects is quite important in the current scenario. For enhancing the quality of the fabric, this paper proposes a Texture Defect Detection (TDD) algorithm. This TDD algorithm utilizes pre-processing for the extraction of luminance plane and Discrete wavelet frame decomposition for dividing the image into several subbands with same resolution as input image. Statistical features are extracted using Gray Level Co-occurrence Matrix and these features are applied to Support Vector Machine for classifying the defective images. This improves the quality of texture segmentation and classification of visible defects. The experimental setup is done with the fabric conveyor and three high resolution industrial cameras acA4600-7gc for covering the entire width of the fabric while running. This TDD algorithm is developed under LabVIEW platform. Textile Texture Database (TILDA) multi-class dataset is used for testing the proposed algorithm. This algorithm is tested for 4 different classes of fabric defects including 2800 defective and 400 non defective fabric images. The success rate of detection of fabric defect is 96.56% with the images from the database. The validation results with real time fabric images show 97% of accuracy in the detection of defects in fabric images.

Keywords Fabric defects, Texture defect detection · TILDA · Classifiers · LabVIEW

1 Introduction

The textile industry is the most important source of income and expenditure for many countries because of the basis for many consumables such as clothing, bags, furniture, and covers. Quality management in the textile industries is of vital importance because it can

✉ T. Meeradevi
tmeeradevi@gmail.com

¹ Electronics and Communication Engineering, Kongu Engineering College, Perundurai, India

avoid significant financial losses. The traditional quality control processes depend mainly on human intervention, but manual defect labeling systems detect 45–65 percent of fabric defects. Therefore, automated defect detection systems are vital to reduce costs and to speed up the quality control process [1].

Detection of fabric defects in textile manufacturing industries is much essential. The traditional approach of quality control uses manual examination. This has to be done at the front of the machine, monitoring the fabric continually and resolving the issues during detection. Although the quality control inspector is properly qualified and experienced, the detection is likely to be limited in accuracy, consistency and efficiency in spotting the problems. The quality control inspector can be tired or bored, and hence uncertain and partial outcomes can be expected. In automated inspection, the challenge of detecting minor faults that locally breaks the homogeneity of texture patterns is resolved and various kinds of defects are classified. Different strategies for fabric defect inspection have been proposed, and such strategies are based on statistics, spectral, model learning, structural and hybrid approaches. Both frequency and spatial information is needed for fabric defect detection. Frequency information is required for fabric image recognition, and spatial information is needed for identifying the position of defect [1].

Quality assurance (QA) in manufacturing is vital to ensure products meet standards, yet manual QA processes are expensive and slow. Artificial Intelligence (AI) offers an appealing solution for automation and expert assistance. Convolutional Neural Networks (CNNs) are increasingly popular for visual inspection tasks. These networks excel in analyzing visual data, making them valuable for QA in manufacturing.

Explainable Artificial Intelligence (XAI) systems complement AI methods by providing transparency and interpretability. They offer insights into how AI makes decisions, crucial for quality inspections in manufacturing. XAI systems aid in understanding AI's decision-making processes, ensuring confidence in automated inspections while maintaining transparency and interpretability, vital for QA in manufacturing.

Santhosh et al. (2020) [2] have described the ways to use a CNN to identify fabric defects. The repeated texture in the fabric is calculated by the autocorrelation value of a fabric image [2].

Meenakshi Garg et al. (2020) [3], H. Li et al. (2020) [4] have proposed the CNN based fabric defect detection. Deep learning approach is proposed for the detection of fabric defects using CNN. Here minimization of the mean square error was used because of its best performance.

Subrata Das et al. (2020) have developed an artificial feed forward neural network based defect detection. The system consists of two parts: image feature vectors and performance assessment of the attributes using a neural classifier to find faults [5].

Chetan Chaudhari et al. (2020) [6] have proposed the use of wavelet transform for defect detection. The wavelet transform is applied on the input image to extract the approximate sub-image of appropriate level. The energy of the sub images is calculated using Parseval's theorem. The energy that deviates from the threshold value contains defects in the corresponding input image.

Chang et al. (2018) have presented a templatebased correction technique for the identification of fault. A fabric image is split into arrays by regularity and the impact of mismatch between arrays is also minimized. Non-defect image arrays are selected for a consistent reference to an average template [7].

Gharsallah et al. (2020) [8], have developed a fabric defect detection method based on the filtering technique. A filter combined with image features is proposed and the threshold value is used to isolate the defects.

A new method based on deep fusion and non-convex called regulated Robust Principal Component Analysis (RPCA) is proposed by Yan Dong et al. (2020) to detect fabric defects. The study extracts deep multilevel functions to distinguish between complicated and diversified textile defects and the RPCA separates the defects from the backdrop. To improve the outcome of detection, a new RPCA based fusion approach is implemented [9].

In the same way, a fault detection method has been developed by Zhang et al. (2020) by using the saliency analysis of the Local Steering Kernel (LSK). The study converts an RGB image of the fabric into the colour space of Commission International Eclairage (CIE) L^*a^*b and then calculates LSK for the singular decomposition value in each colour channel. The cosine matrix correspondence of the desired defective maps is used for measuring similarity among different LSK features. Finally, a multi-scale average fusion system is applied to integrate the defective maps in the final defective map in different scales [10].

Khowaja et al. (2019) have proposed a fabric defect detection using histogram techniques. In this, the defective fabric image is converted into a grayscale image and a well-defined threshold function is utilized to find the fault from histogram [11].

The fabric defect detection problem is resolved under complex lighting conditions by Huang Wang et al. (2020). Recurrent Attention Model (RAM) is used to isolate defects, which is insensible to light and noise differences [12].

Liu et al., (2021) The study presents an upgraded YOLOv4 algorithm for fabric defect detection, integrating a novel SoftPool-based SPP structure that enhances accuracy by 6% while incurring only a 2% decrease in FPS. Employing contrast-limited adaptive histogram equalization for image enhancement fortifies the model against interference, demonstrating improved defect localization precision and speed. [24]. Jia et al. (2022) An advanced fabric defect detection system is introduced, employing transfer learning and an enhanced Faster R-CNN. By utilizing pre-trained Imagenet weights, integrating ResNet50 and ROI Align, and combining RPN with FPN, the system significantly improves detection accuracy, convergence, and identification of small target defects. The cascaded modules and varied IoU thresholds further enhance sample distinction, demonstrating superior performance compared to existing models and offering valuable insights for future fabric defect detection methodologies.. The study aims to fine-tune pre-trained models using adaptive learning techniques for improved accuracy in identifying different types of defects [25]. Alireza Saberironaghi et al., (2023) reviewed various deep learning techniques for defect detection for industrial products. Deep learning-based detection of surface defects on industrial products is discussed from three perspectives: supervised, semi-supervised, and unsupervised. Also, the common challenges and its solution for defect detection with respect to real time images were discussed [26].

In this paper, by considering the need for textile industries at Tirupur and Erode districts, this TDD algorithm is proposed. This camera setup may be fixed on the top of the fabric conveyor and may be connected with the proposed system, so that this fabric defect detection process will be automated and it will reduce the manpower and the error rate. This TDD algorithm detects defects in a texture based on a texture classifier trained with texture samples without any defect. During inspection, the algorithm identifies the defective regions that do not match with the trained defect-free texture samples. The defects identified appear in the output image as blobs. The particle analysis tools in the National Instruments Vision library are utilized to analyze the properties of the defects detected.

The other sections of the paper are organized as follows. Section 2 details the proposed model with TDD algorithm. Section 3 describes the experimental setup of the proposed algorithm. Section 4 the results are analyzed with TILDA dataset and experimental dataset and Section 5 concludes the paper with future scope.

2 Proposed methodology

In general, the running fabric with may be around 1.5 m to 1.7 m. Each camera cover 0.65 m width of the fabric. So in our experimental setup three cameras are fixed to cover the entire width of the fabric with 0.1 m overlapping. These three real time images are sent to the proposed TDD algorithm which is given in Fig. 1.

2.1 TDD algorithm

The step-by-step procedure of the TDD algorithm is as follows.

Step 1: Acquire Image

Fabric images are acquired using a real time camera with the resolution of 4608×3288 or from TILDA dataset (768×512).

Step 2: Pre-Process the Image

The TDD algorithm processes the gray scale image so that if the input image is a colour image, then its single plane (Red plane or Green plane or Blue plane) will be extracted.

Step 3: Locating Fault Area

The geometric pattern matching algorithm finds Region of Interest to find the fault area.

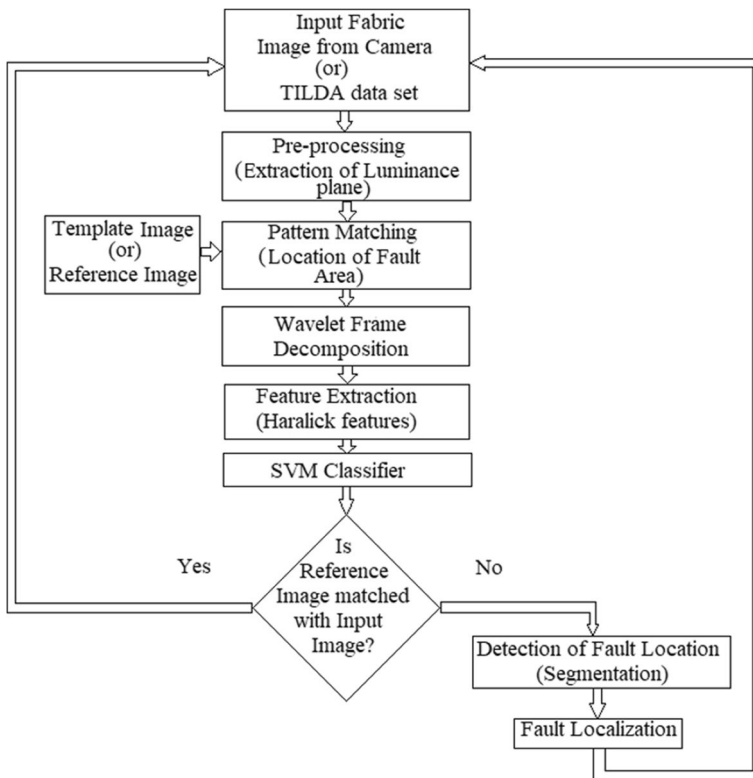


Fig. 1 Texture Defect Detection Algorithm

Step 4: Discrete Wavelet Frame Decomposition.

- The defective image is divided into several subbands using wavelet frames [13]
- The subband image is of the same resolution as the input image and this improves the texture classification and the segmentation capability

Step 5: Statistical Feature Extraction

- TDD algorithm considers second order statistics for feature extraction
- The gray-level co-occurrence matrix (GLCM) is used to create the second-order statistical parameters
- The algorithm divides each subband image into non-overlapping windows and evaluates the coefficient distribution of each window using GLCM $[I(x, y)]$
- The GLCM calculates the probability of a pixel value that occurs at a distance vector \vec{d} from another pixel value
- A texture image $I(x, y)$ is an $N \times M$ matrix that consists of G different grey shades and the displacement vector $\vec{d} = (d_x, d_y)$ is a $G \times G$ matrix as given in the Eq. (1):

$$P_{\vec{d}}(i, j) = \sum_{x=1}^N \sum_{y=1}^M \delta \left\{ I(x, y) = i^{I(x+d_x, y+d_y)} = j \right\} \tag{1}$$

where $\delta\{true\} = 1$ and $\delta\{false\} = 0$.

- The number in the element (i, j) of the GLCM matrix $P_{\vec{d}}(i, j)$ indicates the number of times the pixel level i occurs at the displacement vector \vec{d} from pixel level j .
- The TDD extracts the five Haralick features such as entropy, dissimilarity, contrast, homogeneity and correlation from the GLCM calculated at each partition of the subband texture given in Eq. (2) to (6) [25].

$$Entropy = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} (-\ln P_{i,j}) \tag{2}$$

$$Dissimilarity = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} |i - j|^2 \tag{3}$$

$$Contrast = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} (i - j)^2 \tag{4}$$

$$Homogeneity = \sum_{i=1}^G \sum_{j=1}^G \frac{P_{i,j}}{1 + (i - j)^2} \tag{5}$$

$$Correlation = \sum_{i=1}^G \sum_{j=1}^G P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{\sqrt{\sigma_i^2 \sigma_j^2}} \right] \tag{6}$$

Where,

$\mu_i = \sum_{j=1}^G i P_{i,j}$ and $\mu_j = \sum_{i=1}^G j P_{i,j}$ are the mean values of GLCM.

$\sigma_i = \sum^G P(1 - \mu_i)^2$ and $\sigma_j = \sum^G P(1 - \mu_j)^2$ are the variances of GLCM [14].

Step 6: Support Vector Machine Classifier

- SVM classifiers find a separating or a hyperplane surface that is positioned as far as it is feasible in any one of the two classes from the closest data point
- The classifier considers the spatial distribution information for each sample to determine whether the sample belongs to the known class or not
- The objective of training is to reduce the error function:

$$\min_{w, b, \xi} = \frac{1}{2} W^T W - \rho + \frac{1}{v_l} \sum_{i=1}^l \xi_i \quad (7)$$

Subject to $W^T K(X_i) \geq \rho - \xi_i$; $\xi_i \geq 0, i = 1 \dots l$; $\rho \geq 0$.
where,

W is the hyperplane's normal vector to the origin, $v(nu)$ is the parameter for the upper bound and lower bound of error and vector classes and ξ is the slack variable [15].

Step 7: If there is no defect in the texture image, go to Step 10

Step 8: If there is a defect in the texture image, segment the defect in the texture image

Step 9: Indicate the defect area in the texture image and go to Step 10

Step 10: Continue the same process with the next texture image

2.2 Dataset

This proposed algorithm is initially tested with the TILDA dataset which is a standard dataset for fabric defects. After validating this algorithm, it tested with the real time captured data. In TILDA dataset four primary classes (c1-c4) are given and are aligned according to the surface structure. For every class, two representative (r1 & r2 or r2 & r3 or r1 & r3) subgroups are included. The details of this dataset's main class are given in Table 1. Each subgroup contains 50 faultless images (e0) and seven error classes (e1-e7). The images are in Tag Image File Format (TIFF) and have a dimension of 768×512 pixels. For each image, an accompanying text with a brief description of the Error (location and size) is provided in the dataset itself. The TILDA dataset contains a total of 3,200 images, 2,800 text reads with error descriptions and a data volume of 1.2 Giga Bytes [16]. Figure 2 depicts the directory structure of the dataset. Table 2 describes each sub-error class and the type of fault in the main class. The images in the dataset are designated to the CREN.tif convention and are described in Table 3.

For example, the image in the TILDA dataset c3r1e0n40.tif represents the class 3 (c3) with the representative subgroup 1 (r1), the error class 0 (e0) and the 40th image [16].

In this paper, totally, 3200 images are used for training and testing purposes with the error class from e0–e7 from the TILDA dataset which is described in Table 4. The sample images from the four main classes of the TILDA dataset are provided in Fig. 3.

Table 1 Description of the main class and the type of fabrics [16]

| Type of class | Type of fabric | Example |
|---------------|--|---|
| Class 1(c1) | Internal structure that is either very fine or not is visible | Silk or viscose that has not been printed |
| Class 2(c2) | Fabrics with a stochastic structure of low variance and the surface do not contain any imprint | Wool or jute |
| Class 3(c3) | Fabrics with periodic pattern | Curtains or materials that have a diamond pattern printed on them |
| Class 4(c4) | Fabrics printed with no discernible patterns | Flowers of various sizes are imprinted on viscose |

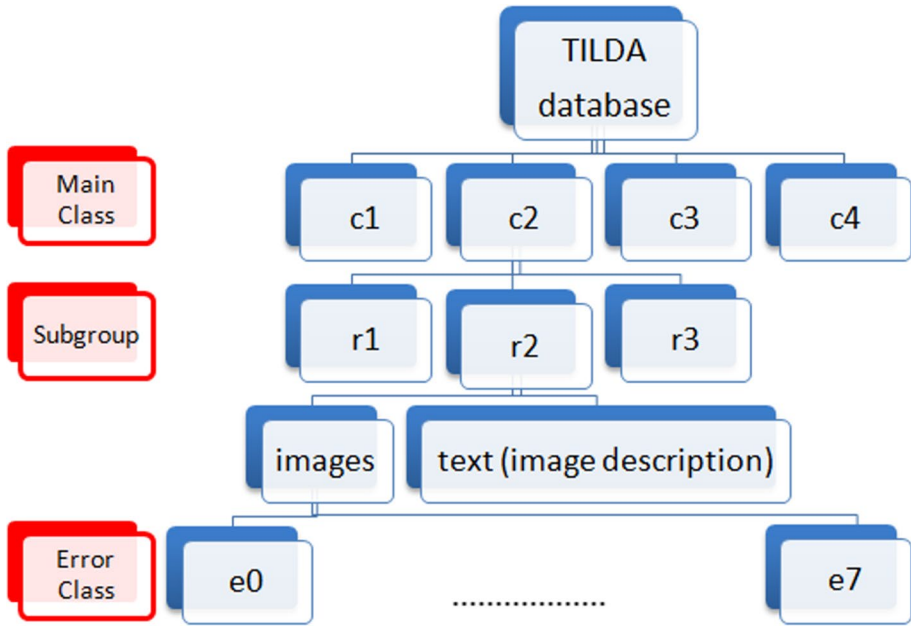


Fig. 2 Directory structure of the TILDA dataset

Table 2 Description of the error class and the type of fault Zhang et al.,(2015)

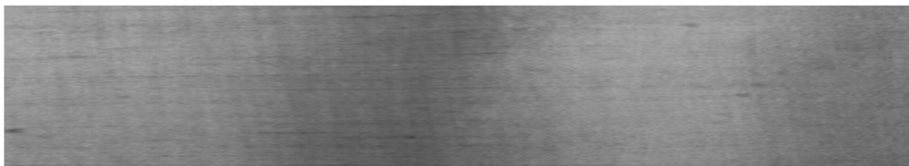
| Error Class | Type of fault |
|-------------|---|
| e0 | No defects in the fabric |
| e1 | Mechanical grievance causes holes and cuts in the material |
| e2 | Oil corners and color defects |
| e3 | Thread error, compaction of threads (without mechanically caused cracks). absence of individual threads in the fabric |
| e4 | Foreign bodies on the tissue (called flight) |
| e5 | Wrinkles in the tissue (without mechanical damage) |
| e6 | Lighting has been changed |
| e7 | A distortion from the tilting of the camera and changes in the distance from the camera to the test |

Table 3 Conventional symbols

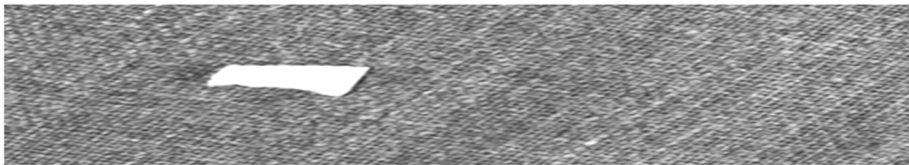
| Conventional symbol | Explanation |
|---------------------|---|
| C | Indicates the main class number |
| R | Indicates the representative subgroup of the class number |
| E | Indicates the error class number |
| N | Indicates the image number |

Table 4 Number of images in TILDA data Zhang et. al., (2015)

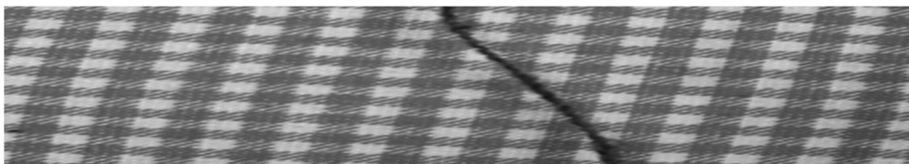
| Main Class | Representative Subgroups | Error Class | Number of images |
|------------|--------------------------|-------------|---------------------|
| c1 | r1 | e0-e7 | $8 \times 50 = 400$ |
| | r3 | e0-e7 | $8 \times 50 = 400$ |
| c2 | r2 | e0-e7 | $8 \times 50 = 400$ |
| | r3 | e0-e7 | $8 \times 50 = 400$ |
| c3 | r1 | e0-e7 | $8 \times 50 = 400$ |
| | r3 | e0-e7 | $8 \times 50 = 400$ |
| c4 | r1 | e0-e7 | $8 \times 50 = 400$ |
| | r3 | e0-e7 | $8 \times 50 = 400$ |
| Total | | | 3200 |



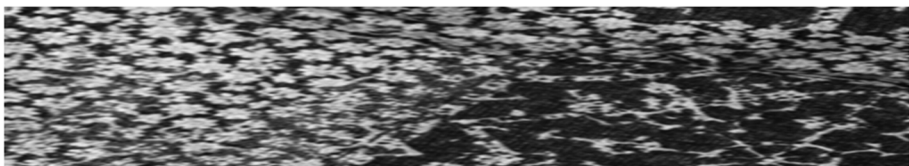
(a) c1r1e5n22.tif



(b) c2r2e4n27.tif



(c) c3r1e4n4.tif



(d) c4r1e5n29.tif

Fig. 3 Sample images from c1, c2, c3 and c4 classes with the corresponding image name given in TILDA data set. (a) image with a single fold across the fabric. (b) a medium-sized piece of paper on the top of the fabric. (c) image with single thread across the fabric and (d) image with a twofold across the fabric

3 Experimental setup

The experimental setup is shown in Fig. 4 for the proposed detection of fabric defect and shade variation using TDD algorithm. Experiments are conducted using LabVIEW software, an Industrial Controller 3173—1P20 with the Intel Core i7-5650U @ 2.2 GHz and Xilinx Kintex-7 XC7K160T Processor. Three acA4600-7gc Basler cameras are connected to capture the image with the resolution of 4608×3288 . They are connected to the controller using Ethernet cable and powered by a basler power cable.

3.1 Working of the experimental setup

The working table consists of a motor with roller which moves the fabric continuously with a constant speed on the working table. The images of the fabric are captured by the three Basler Industrial cameras with 7 frames per second. The three cameras cover the whole breadth (1.5 m) of the fabric. Then the images captured are fed continuously to the industrial controller which processes them using the TDD algorithm. The defects of the fabric detected are continuously displayed in the monitor.

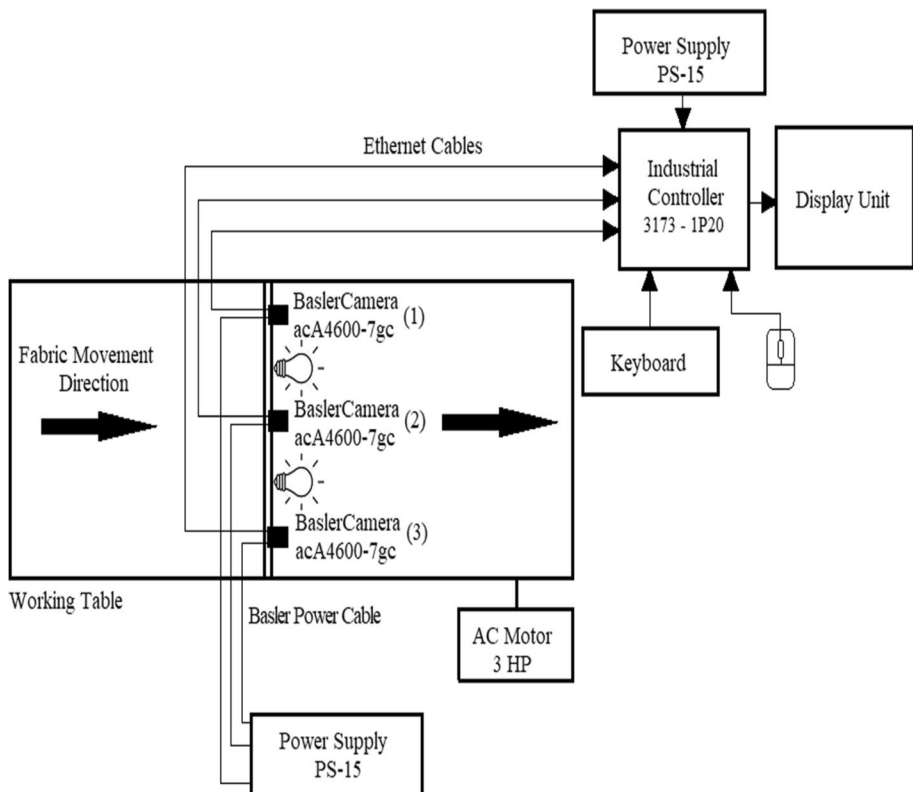


Fig. 4 Experimental setup for the detection of faults in fabrics

4 Results and discussion

4.1 Results for TILDA dataset

The confusion matrix is calculated by applying the TDD algorithm on TILDA dataset.

The performance parameters utilized are:

Sensitivity It is the accurate determination of bad samples. It is also known as a recall it is represented in Eq. (8).

$$\text{Sensitivity } (S_E) = \frac{TP}{TP + FN} \quad (8)$$

Specificity It is the proper detection of non-defect samples as shown in Eq. (9):

$$\text{Specificity } (S_p) = \frac{TN}{TN + FP} \quad (9)$$

Detection Success Rate (DSR) It demonstrates how the model anticipates proper outputs on a consistent basis. It is referred to as Detection Accuracy (DA) and can be calculated using the Eq. (10):

$$\text{Detection Accuracy (DA)} = \text{DSR} = \frac{\text{Total No. of valid prediction by the classifier}}{\text{Total No. of prediction by the classifier}}$$

$$\text{DSR} = \frac{TN + TP}{TN + FP + FN + TP} \quad (10)$$

False Alarm Rate (FAR) It is likely that a false alarm will be raised if the true value is negative. It can be calculated using Eq. (11):

$$\text{False Alarm Rate (FAR)} = \frac{\text{No. of samples detected to be defective}}{\text{No. of defect-free image samples}}$$

$$\text{FAR} = \frac{FP}{FP + FN} \quad (11)$$

Detection Rate (DR) This is a result of the model that predicts the positive class accurately. Equation (12) can be used to compute DR:

$$\text{Detection Rate (DR)} = \frac{\text{No. of defective image samples detected correctly}}{\text{No. of defective image samples}}$$

$$\text{DR} = \frac{TP}{TP + FN} \quad (12)$$

Positive Predictive Value (PPV) or Precision It gives the number of correct outputs out of all the correctly predicted positive values by using the TDD algorithm. It determines whether an algorithm is reliable or not. For calculating the precision, Eq. (13) is utilized [17].

$$\text{Positive Predictive Value (PPV) or Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (13)$$

Totally, 50 images from the error free e0 of c1 and r1 class and 350 images from the error class e1-e7 from the same c1 and r1 class are tested and the confusion matrix is obtained. The same process is repeated for the remaining classes c2-c4 and the confusion matrix is given in Table 5.

The validation results of c1 are shown in Fig. 5 and that of c3 are shown in the Fig. 6. The Figs. 5 and 6 shows the incorrect results as 18/400 and 9/400 respectively. The parameters are evaluated using Eqs. (8) to (13) and shown in Table 5 for the TILDA dataset.

Furthermore, Table 6 depicts the highest accuracy scores of 95.50%, 96.25%, 97.75% and 98% for c1, c2, c3 and c4 texture class respectively based on TILDA dataset using the proposed algorithm with SVM classifier. The average accuracy of 96.65% is achieved by using the algorithm. The defect detected image with the source images of c1 and c2 classes are shown in Fig. 7. The defect detected is covered with the red outline.

The TDD algorithm works under LabVIEW software environment which is a new platform utilized for detection of the fabric defect. The comparison of the existing defect detection techniques with the proposed model is given in Table 7. In most of these studies, self-made datasets have been employed, but TILDA, which has a public and extensive database, has been the most regularly used. Table 7 compares the accuracy rates of the proposed model with the TILDA based investigations. Based on the volume of classes and images, it is observed from the results that the proposed algorithm is better than the existing ones.

In addition, the proposed approach clearly appears to be the most comprehensive study using all classes and images in the TILDA dataset. In addition to all these, Zhang et al. (2015), Salem and Nasri (2011), Salem and Abdelkrim (2020), and Deotale and Sarode

Table 5 Confusion matrix for TILDA dataset

| Main Class | Representative Subgroups | Number of Error Class Images | Confusion Matrix for 400 images |
|------------|--------------------------|------------------------------|--|
| c1 | r1 | e0=50 e1-e7=350 | $\begin{bmatrix} 333 & 1 \\ 17 & 49 \end{bmatrix}$ |
| | r3 | e0=50 e1-e7=350 | $\begin{bmatrix} 343 & 12 \\ 7 & 38 \end{bmatrix}$ |
| c2 | r2 | e0=50 e1-e7=350 | $\begin{bmatrix} 344 & 10 \\ 6 & 40 \end{bmatrix}$ |
| | r3 | e0=50 e1-e7=350 | $\begin{bmatrix} 342 & 7 \\ 8 & 43 \end{bmatrix}$ |
| c3 | r1 | e0=50 e1-e7=350 | $\begin{bmatrix} 349 & 8 \\ 1 & 42 \end{bmatrix}$ |
| | r3 | e0=50 e1-e7=350 | $\begin{bmatrix} 344 & 9 \\ 6 & 41 \end{bmatrix}$ |
| c4 | r1 | e0=50 e1-e7=350 | $\begin{bmatrix} 342 & 2 \\ 8 & 48 \end{bmatrix}$ |
| | r3 | e0=50 e1-e7=350 | $\begin{bmatrix} 345 & 3 \\ 5 & 47 \end{bmatrix}$ |

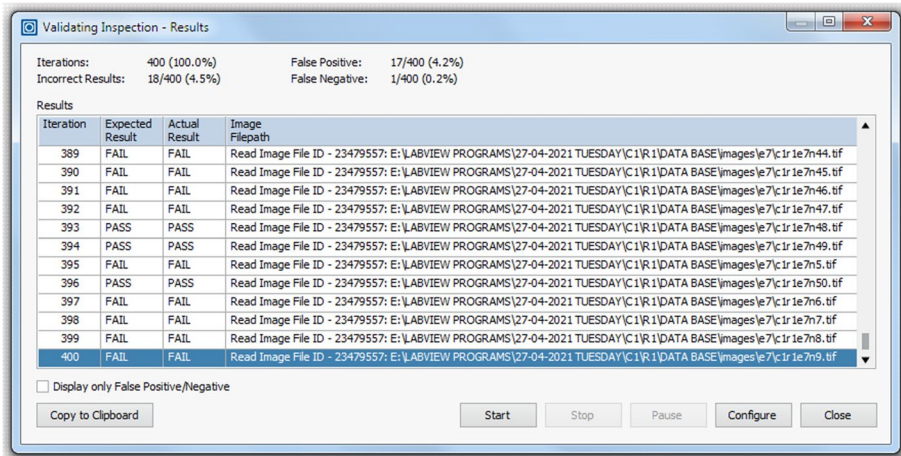


Fig. 5 Validation results of class c1 with subgroup r1

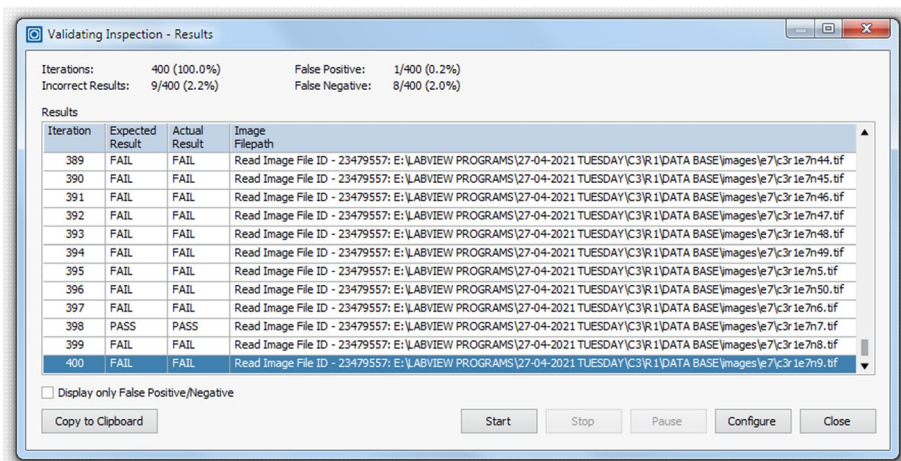
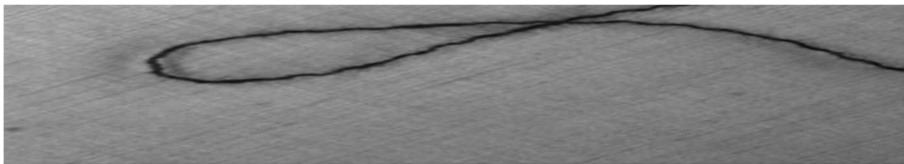


Fig. 6 Validation results of class c3 with subgroup r1

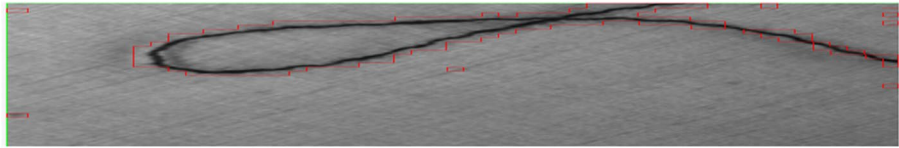
(2019) from previous studies are used traditional feature extraction such as (GLCM, LBP, etc.) and classifier (SVM, Neural Network, etc.) methods for detection of texture defect. The highest accuracy among these studies is achieved at 97.6% by Zhang et al. On the other hand, Jeyaraj and Nadar (2019), Jeyaraj and Nadar (2020), and Jing et al. (2019) are used pre-trained deep models such as AlexNet, ResNet512, and AlexNet, respectively. The highest accuracy among these studies is achieved at 98.5% by Jeyaraj and Nadar (2020)). According to all these results, it has been determined that TDD algorithm are more successful than conventional methods, considering the number of classes and images. In each of these other studies that used the TILDA dataset, a certain number of classes and images are used rather than opting to use all classes and texture images. However, the proposed model, which is tested using all classes and images from the TILDA dataset, and achieved a superior level of success when compared to the existing studies based on fewer classes and images.

Table 6 Performance parameters for TDD algorithm

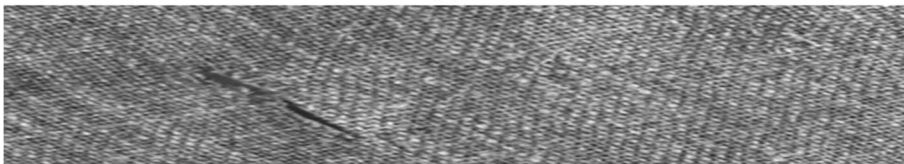
| Main Class | Representative Subgroups | SE | SP | DA | FAR | DR | PPV |
|---------------|--------------------------|------|------|--------|--------|------|------|
| c1 | r1 | 0.87 | 0.74 | 0.9550 | 0.2576 | 0.13 | 0.95 |
| | r3 | 0.90 | 0.84 | 0.9525 | 0.1556 | 0.10 | 0.98 |
| c2 | r2 | 0.90 | 0.87 | 0.9600 | 0.1304 | 0.10 | 0.98 |
| | r3 | 0.89 | 0.84 | 0.9625 | 0.1569 | 0.11 | 0.98 |
| c3 | r1 | 0.89 | 0.98 | 0.9775 | 0.0233 | 0.11 | 1.00 |
| | r3 | 0.89 | 0.87 | 0.9625 | 0.1277 | 0.11 | 0.98 |
| c4 | r1 | 0.88 | 0.86 | 0.9750 | 0.1429 | 0.12 | 0.98 |
| | r3 | 0.88 | 0.90 | 0.9800 | 0.0962 | 0.12 | 0.99 |
| Average Value | | 0.89 | 0.86 | 0.9656 | 0.1363 | 0.11 | 0.98 |



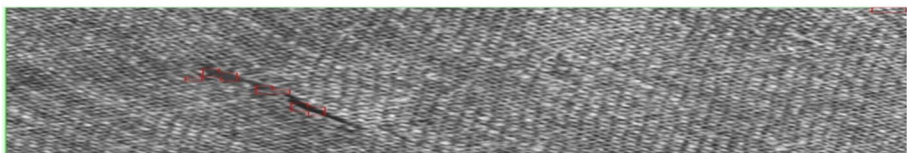
(a)



(b)



(c)



(d)

Fig. 7 Defect detected image with the source image of Class c1 and c2 (a) the source image c1r1e4n11.tif from Class c1 (b) the defect detected c1r1e4n11.tif image with the defect marked in red line (c) the source image c2r2e1n18.tif from Class c2 and (d) the defect detected c2r2e1n18.tif image with the defect marked in red

Table 7 Comparison of the results of proposed model with the previous models

| References | Methods | Data Set | Number of error class and Images | Accuracy Score |
|------------------------------|---|--|----------------------------------|-----------------|
| Jing et al., 2019 [18] | Fine-tuned AlexNet | C1-R1 | 6 error class and Images | 97.2% |
| Zhang et al., 2015 [19] | LBP, GLCM and Neural Network | C1-R1 and C1-R2 | 6 error class and 600 Images | 97.6% |
| Salem & Nasri, 2009 [20] | LBP, GLCM and SVM | - | 7 error class and 480 Images | 86.7% |
| Salem & Abdelkrim, 2020 [21] | GLCM, LBP, LPQ, and SVM | - | 5 error class and 480 Images | 97.25% |
| Jeyaraj & Nadar, 2020 [22] | ResNet512 based CNN features, Kullback Leibler Divergence (KLD) and Markov Random Field (MRF) | C1-R2, C2-R2 | 6 error class | 98.5% and 96.5% |
| Jeyaraj & Nadar, 2019 [23] | AlexNet based Multi-scaling deep CNN | - | 6 error class and 1850 images | 96.55% |
| Deotale & Sarode, 2019 [24] | GLCM, Gabor Wavelet, and Random Decision Forest | - | 6 error class | 84.5% |
| Proposed Model | TDD Algorithm | C1 (R1, R3), C2 (R2, R3) C3 (R1, R3), and C4 (R1, R3) | 8 error class and 3200 image | 96.56% |

4.2 Results of experimental setup

The validated TDD algorithm with TILDA dataset is applied to the real experiment and the performance is reported. The working model of the fabric defect detection with the industrial setup and the controller assembly is shown in Fig. 8.

Totally, 40 error free textile images and 60 error images are tested in real time and confusion matrix is plotted and shown in Table 8. The average accuracy of 97% is achieved using the proposed TDD algorithm.

A sample defect identified image is shown in Fig. 9. The inspection interface of the LabVIEW environment contains three main areas: Results panel, Inspection Statistics panel and Display window.

The shade variation value and the maximum defect detection area are 69 & 1.6 mm² respectively. Results panel list the steps in the inspection by name. For each step in the inspection, it displays the step type and results (PASS or FAIL). Likewise, the status of PASS in shade variation should have the histogram value of 99 to 100. The status of PASS in the maximum defect detection area should have the value of 0.0 mm². For Fig. 9(a) it is PASS and for Figs. 9(b) & 9(c) it is FAIL. The display window displays the image under inspection and the status of the defect. For Figs. 9 a,b and c the defected fabric images with defect indicated in red outline are displayed. The shade variation value and the maximum defect detection area values are also displayed. Inspection statistics panel contains the processing time of the inspection. From Figs. 9 (a, b and c), it is 0.24 ms, 0.23 ms, and 0.23 ms respectively. So, the average time of inspection is 0.23 ms for detecting the defect. Thus, the TDD algorithm with LabVIEW environment efficiently detects the defects in the fabric.

Fig. 8 Working model for fabric defect detection with industrial setup (a) Industrial Controller 3173—1P20 (b) acA4600-7gc Basler camera and (c) Fabric Conveyor

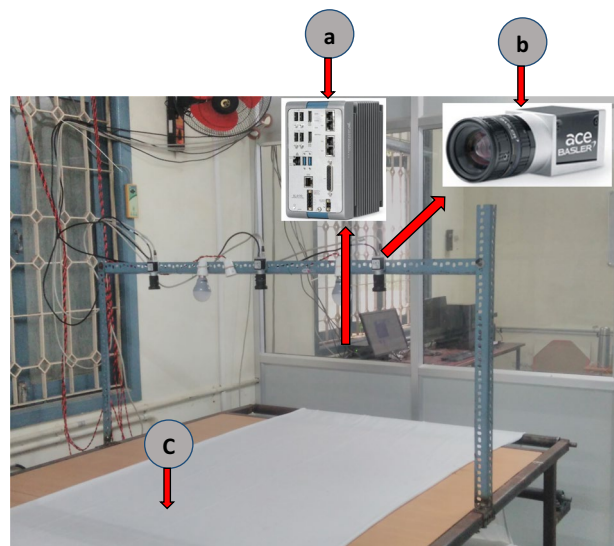
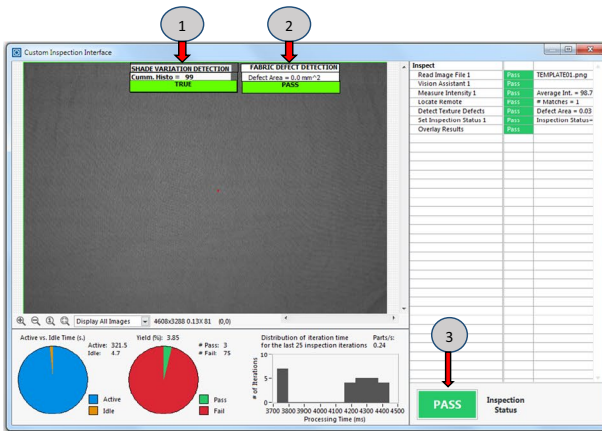


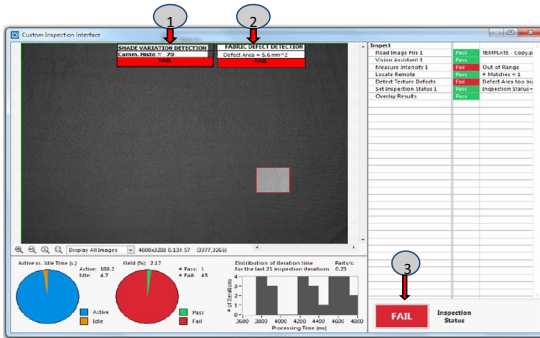
Table 8 Confusion matrix for the real time fabric defect detection

| Name of the defect | Confusion Matrix | SE | SP | DA | FAR | DR | PPV |
|--|--|-------|------|--------|--------|------|------|
| Holes and cuts | $\begin{bmatrix} 58 & 1 \\ 2 & 39 \end{bmatrix}$ | 0.60 | 0.95 | 0.9700 | 0.0488 | 0.40 | 0.97 |
| Oil corners and color defects | $\begin{bmatrix} 57 & 1 \\ 3 & 39 \end{bmatrix}$ | 0.59 | 0.93 | 0.9600 | 0.0714 | 0.41 | 0.95 |
| Thread error and absence of individual threads in the fabric | $\begin{bmatrix} 58 & 1 \\ 2 & 39 \end{bmatrix}$ | 0.60 | 0.95 | 0.9700 | 0.0488 | 0.40 | 0.97 |
| Foreign bodies on the fabric (called flight) | $\begin{bmatrix} 59 & 2 \\ 1 & 38 \end{bmatrix}$ | 0.61 | 0.97 | 0.9700 | 0.0256 | 0.39 | 0.98 |
| Wrinkles on the fabric (without mechanical damage) | $\begin{bmatrix} 59 & 2 \\ 1 & 38 \end{bmatrix}$ | 38.00 | 0.97 | 0.9700 | 0.0256 | 0.39 | 0.98 |
| Changed lighting conditions | $\begin{bmatrix} 58 & 1 \\ 2 & 39 \end{bmatrix}$ | 0.60 | 0.95 | 0.9700 | 0.0488 | 0.40 | 0.97 |
| A distortion from the tilting of the camera | $\begin{bmatrix} 59 & 1 \\ 1 & 39 \end{bmatrix}$ | 0.60 | 0.98 | 0.9800 | 0.0250 | 0.40 | 0.98 |
| Average Value | | 0.60 | 0.96 | 0.9700 | 0.0420 | 0.40 | 0.97 |



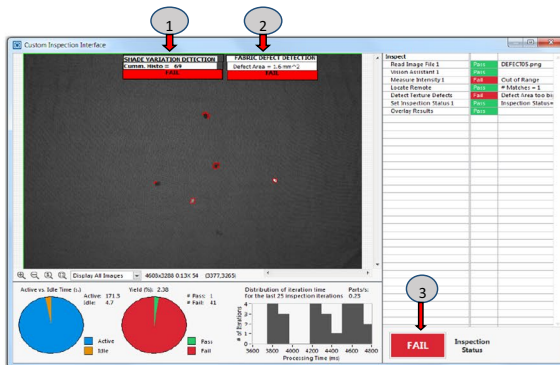
1. PASS status indicates that there is no shade variation in the fabric (Histogram = 99)
2. PASS status indicates that there is no defect in the fabric (Max. Defective area = 0.0 mm²)
3. PASS status for the fabric image with no defect

(a)



1. FAIL status indicates the shade variation in the fabric (Histogram = 70)
2. FAIL status indicates the defect in the fabric (Max. Defective area = 5.6 mm²)
3. FAIL status for the fabric image with defect

(b)



1. FAIL status indicates the shade variation in the fabric (Histogram = 69)
2. FAIL status indicates the defect in the fabric (Max. Defective area = 1.6 mm²)
3. FAIL status for the fabric image with defect

(c)

◀ **Fig. 9** Defect identified images for the experimental setup (a) the image with no defect, the shade variation value and the maximum defect detection area value of 99 & 0.0 mm² respectively, (b) the image with defect. The shade variation value and the maximum defect detection area value are 70 & 5.6 mm² respectively and (c) the image with defect

5 Conclusion and future work

In this paper, a SVM classifier based on TDD algorithm with LabVIEW environment is proposed to characterize visual fabric defect detection. The TDD algorithm converts the defect image into gray scale image, and then the image is segmented using wavelet decomposition method. Haralick features are extracted using GLCM method. The performance is calculated using the SVM classifier. To validate the TDD algorithm, the multi-class TILDA dataset is employed. Comprehensive validation results are 95.50%, 96.25%, 97.75% and 98% for c1, c2, c3, and c4 class respectively. Furthermore, the TDD algorithm has the mean overall precision score of 96.56% for all the four classes. This proposed work is limited to the running fabrics with or without print on the entire fabric with four main classes of fabric defects. Future work may be extended for all types of defects and all types of fabrics including T-shirt printing.

Data availability All the details about data availability are mentioned within this manuscript.

Declarations

Conflict of interest Authors declare no conflict of interest. This project is sanctioned by Department of Science and Technology-State Science and Technology (DST-SSTP) Programme (DST/SSTP/2018/232).

References

1. Karlekar VV, Biradar MS, Bhangale KB (2015) Fabric defect detection using wavelet filter. In: 2015 International Conference on Computing Communication Control and Automation. ICCUBEA, Pune, India, pp 712–715. <https://doi.org/10.1109/ICCUBEA.2015.145>
2. Santhosh KK, Tamil SR, Uthaya KM, Jaya VP, Finney DS (2020) Defect detection in fabrics using modified CNN. *Waffen-Und Kostumkunde Journal*, XI(VI) 233–236. <https://doi.org/10.11205/WJ.2020.V11I6.05.100937>
3. Garg M, Dhiman G (2020) Deep convolution neural network approach for defect inspection of textured surfaces. *J Inst Electron Comput* 2:28–38. <https://doi.org/10.33969/JIEC.2020.21003>
4. Li H, Zhang H, Liu L, Zhong H, Wang Y, Jonathan Wu QM (2020) Integrating deformable convolution and pyramid network in Cascade R-CNN for fabric defect detection. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Toronto, ON, Canada, pp 3029–3036. <https://doi.org/10.1109/SMC42975.2020.9282875>
5. Das S, Wahi A, Keerthika S, Thulasiram N (2020) Defect Analysis of Textiles Using Artificial Neural Network. *Curr Trends Fashion Technol Textile Eng* 6(1):01–05. <https://doi.org/10.19080/CTFTTE.2020.06.555677>
6. Chaudhari C, Gupta RK, Fegade S (2020) A hybrid method of textile defect SVD and wavelet transform. *Int J Recent Technol Eng* 8(6):5356–5360. <https://doi.org/10.35940/ijrte.F9569.038620>
7. XingzhiChang CG, Liang J, Xin X (2018) Fabric defect detection based on pattern template correction Hindawi. *Math Prob Eng* 2018(3709821):18. <https://doi.org/10.1155/2018/3709821>
8. Gharsallah, M. B., & Braiek, E. B. (2020). A visual attention system based on an anisotropic diffusion method for effective textile defect detection. *Journal of Textile Institute*. <https://doi.org/10.1080/00405000.2020.1850613>

9. Dong Y, Wang J, Li C, Liu Z, Xi J, Zhang A (2020) Fusing multilevel deep features for fabric defect detection based NTV-RPCA. *IEEE Access* 8:161872–161883. <https://doi.org/10.1109/ACCESS.2020.3021482>
10. Zhang K, Yan Y, Li P, Jing J, Wang Z, Xiong Z (2020) Fabric defect detection using saliency of multi-scale local steering kernel. *IET Image Processing*, 14(7):1265–1272. *IET Digital Library*. <https://doi.org/10.1049/iet-ipr.2018.5857>
11. Khowaja A, Nadir D (2019) Automatic fabric fault detection using image processing. In: 2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS). Karachi, Pakistan, pp 1–5. <https://doi.org/10.1109/MACS48846.2019.9024776>
12. Wang H, Duan F, Zhou W (2020) Fabric defect detection under complex illumination based on an improved recurrent attention model. *J Textile Inst*. <https://doi.org/10.1080/00405000.2020.1809918>
13. Schulz-Mirbach H (1996) A reference data set for evaluating visual inspection procedures for textile surfaces. Technical University of Hamburg-Harburg, Technical Informatics I, TILDA Textile Texture-Database, Pattern Recognition and Image Processing. Version 1.0
14. Rasheed A, Zafar B, Rasheed A, Ali N, Sajid M, Dar SH, Habib U, Shehryar T, Tariq M (2020) Fabric defect detection using computer vision techniques a comprehensive review. *Math Probl Eng* 2020:01–24. <https://doi.org/10.1155/2020/8189403>
15. Jing J-F, Ma H, Zhang H-H (2019) Automatic fabric defect detection using a deep convolutional neural network. *Color Technol* 135(3):213–223. <https://doi.org/10.1111/cote.12394>
16. Zhang L, Jing J, Zhang H (2015) Fabric Defect Classification Based on LBP and GLCM ★. *J Fiber Bioeng Inform* 8:81–89. <https://doi.org/10.3993/jfbi03201508>
17. Ben Salem Y, Nasri S (2009) Texture classification of woven fabric based on a GLCM method and using multiclass support vector machine. In: 2009 6th International Multi-Conference on Systems, Signals and Devices pp 1–8. <https://doi.org/10.1109/ssd.2009.4956737>
18. Salem YB, Abdelkrim MN (2020) Texture classification of fabric defects using machine learning. *Int J Electr Comput Eng (IJECE)* 10(4):4390–4399. <https://doi.org/10.11591/ijece.v10i4.pp4390-4399>
19. Jeyaraj PR, Nadar ERS (2020) Effective textile quality processing and an accurate inspection system using the advanced deep learning technique. *Text Res J* 90(9–10):971–980. <https://doi.org/10.1177/0040517519884124>
20. Jeyaraj PR, Nadar ERS (2019) Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm. *Int J Clothing Sci Technol* 31(4):510–521. <https://doi.org/10.1108/IJCST-11-2018-0135>
21. Deotale NT, Sarode TK (2019) Fabric defect detection adopting combined GLCM, gabor wavelet features and random decision forest. *Research* 10(1):1–13. <https://doi.org/10.1007/s13319-019-0215-1>
22. Kavin Kumar K, Meera Devi T, Maheswaran S (2018) An efficient method for brain tumor detection using texture features and SVM classifier in MR Images. *Asian Pac J Cancer Prev* 19(10):2789–2794. <https://doi.org/10.22034/APJCP.2018.19.10.2789>
23. ManojSenthil K, Meeradevi T (2017) Performance analysis of feature-based lung tumor detection and classification. *Curr Med Imaging* 13(3):339–347. <https://doi.org/10.2174/1573405612666160725093958>
24. Liu Q, Wang C, Li Y, Gao M, Li J (2022) A Fabric Defect Detection Method Based on Deep Learning. *IEEE Access* 10:4284–4296. <https://doi.org/10.1109/ACCESS.2021.3140118>
25. Jia Z, Shi Z, Quan Z, Shunqi M (2022) Fabric defect detection based on transfer learning and improved Faster R-CNN. *Journal of Engineered Fibers and Fabrics* 17. <https://doi.org/10.1177/15589250221086647>
26. Berironaghi A, Ren J, El-Gindy M (2023) Defect detection methods for industrial products using deep learning techniques: a review. *Algorithms* 16(2):95. <https://doi.org/10.3390/a16020095>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.