Optimal Artificial Neural Network-based Fabric Defect Detection and Classification

Nesamony Sajitha
Department of Computer and Information Science, Faculty of Science, Annamalai University, India
sajithap06@gmail.com (corresponding author)

Srinivasan Prasanna Priya
Thiru A. Govindasamy Govt Arts College, India
prasannapriyatdm@gmail.com

Received: 19 December 2023 | Revised: 4 January 2024 | Accepted: 8 January 2024

 DOI: https://doi.org/10.48084/etasr.6773

ABSTRACT

Automated Fabric Defect (FD) detection plays a crucial role in industrial automation within fabric production. Traditionally, the identification of FDs heavily relies on manual assessment, facilitating prompt repairs of minor defects. However, the efficiency of manual recognition diminishes significantly as labor working hours increase. Consequently, there is a pressing need to introduce an automated analysis method for FD recognition to reduce labor costs, minimize errors, and improve fabric quality. Many researchers have devised defect detection systems utilizing Machine Learning (ML) approaches, enabling swift, accurate, and efficient identification of defects. This study presents the Optimal Artificial Neural Network-based Fabric Defect Detection and Classification (OANN-FDDC) technique. The OANN-FDDC technique exploits handcrafted features with a parameter-tuning strategy for effectively detecting the FD process. To obtain this, the OANN-FDDC technique employs CLAHE and Bilateral Filtering (BF) model-based contrast augmentation and noise removal. Besides, the OANN-FDDC technique extracts shape, texture, and color features. For FD detection, the ANN method is utilized. To improve the detection results of the ANN method, the Root Mean Square Propagation (RMSProp) optimization technique is used for the parameter selection process. The simulation outputs of the OANN-FDDC technique were examined on an open fabric image database. The experimental results of the OANN-FDDC technique implied a better outcome than the 96.97% of other recent approaches.

Keywords-textile industry; fabric defect; machine learning; automation; feature extraction

I. INTRODUCTION

Textile manufacturing is a widespread and complex industrial field [1]. The process of textile manufacturing contains multiple complicated and ordered processes, which mostly comprise dyeing, spinning, printing, weaving, finishing, and manufacturing of garments [2]. The quality and stability of textile fabric manufactured by the entire production lines are essential to some enterprises. FD identification is a quality control process that has to ensure the recognition of defects existing in the textile fabric [3]. These defects can decrease the textile fabric cost by approximately 45-65%. A conventional investigation model is the collection of manual operators and their work is to identify the defects while the machines transfer the fabric [4]. Conventionally, these motored machines unroll the fabric rolls. Consequently, the fabric is stretched and presented to the workers without differences in thickness and folding. This process depends on human attention and visual capability. These tasks can be more time-consuming and tedious which can result in fatigue and human errors [5]. Hence, conventional models usually obtain an accuracy of 60-75%, despite their extremely slow speed when compared to the rate of production. Automated visual investigation models for ensuring the higher quality of products in manufacturing lines are improving requirements [6]. The major benefits of automated defect identification methods are stability, dependability, and higher efficiency [7].

Several researchers have utilized DL methods to FD identification issues and achieved satisfactory outcomes for improving textile production efficiency and product quality [8]. Despite DL techniques have proved to be strong when dealing with classification and segmentation difficulties, there are still a few issues in real-time applications of particular factories [9]. Foremost, the real textile production line needs a higher real-time technique of performance, which is the need for great implementation efficiency [10]. Additionally, when comparing standard defect-free instances, information on defective images is complicated to acquire, which gives challenges to the DL training procedure [11]. Deep Convolutional Neural Network (DCNN)-based techniques have accomplished fitting outputs on visual responsibilities and are broadly applied in the industry.

This study presents the Optimal Artificial Neural Network-based Fabric Defect Detection and Classification (OANN-FDDC) technique. The OANN-FDDC technique exploits handcrafted features with a parameter-tuning strategy for effectively detecting the FD process. To obtain this, the OANN-FDDC technique employs CLAHE and Bilateral Filtering (BF), model-based contrast augmentation, and noise removal. Also, the OANN-FDDC technique extracts shape, texture, and color features. For FD detection, the ANN method was used. To improve the detection results of the ANN method, the Root Mean Square Propagation (RMSProp) optimization technique was used for the parameter selection process. The simulation outputs of the OANN-FDDC technique were examined on a fabric image database in the Kaggle repository.

II. RELATED WORKS

Authors in [12] established the Hybrid Mutation Moth Flame Optimizer with a DL Based Smart FD Detection (HMFODL-FDD) approach to maintainable manufacturing. To realize this purpose, their approach utilized a contrast enhancement procedure for boosting the image quality. For the extracting process, it utilizes the Inceptionv3 approach with an HMFO technique-based hyper-parameter optimizer. The FD classifier utilizes the BPNN approach. Authors in [13] presented a fabric design defect detection approach depended on a vision-based tactile sensor. The robotic arm equipped with the tactile sensor can be utilized for automating and standardizing the data gathered procedure and creating fabric databases. Also, a CNN combined with an attention process from the channel domain was established for detecting fabric types. Authors in [14] examined a learning-dependent structure for the automatic recognition of FDs whereas the Inception-V1 approach was utilized for forecasting the existence of errors in the local regions. Eventually, the authors execute the LeNet-5 approach which roles a vital play in voting, for identifying the FD type. Authors in [15] introduced an extremely effectual DL-based approach for a pixel-level FD classifier based on CNNs. Authors in [16] proposed a CNN-based textile recognition method called Faster R-CNN. Initially, a textile defect-mapping feature is extracted by the ResNet101 DCNNs. Faster R-CNN feature extraction in the last layer of feature mapping generates a loss of low-level place data.

Authors in [17] considered the Attention-Gate based on U-shaped Reconstruction Networking (AGUR-Net) and a dual-threshold segmentation post-processed approach. Authors in [18] introduced the Attention based Feature Fusion GAN (AFFGAN) method for unsupervised recognition of defects of yarn-dyed fabrics. This structure employs an adapted FPN to fuse multilevel data and employs an attention process for enhancing the model feature illustration abilities. This method utilizes a patch-level discriminator and an attention fusion generator. Authors in [19] introduced the FN-Net, a lightweight CNN-based architecture for FD detection. In contrast to the state-of-the-art models, FN-Net demonstrates 3 to 33 times faster training speed with reduced GPU and memory usage. In [20], the presented model utilizes the Bayesian optimization algorithm for network pruning. The training and detection phase utilize the pruned network, followed by employing the image processing flow outlined for the conclusive assessment in FD recognition.

III. THE PROPOSED MODEL

The novel OANN-FDDC technique for the automatic detection and classification of FDs is presented in this study. The OANN-FDDC technique exploits handcrafted features with a parameter-tuning strategy for an effective FD detection process. The OANN-FDDC approach comprises the stages of pre-processing, handcrafted feature extraction, ANN, and RMSProp-based classification and tuning. Figure 1 exhibits the workflow of the OANN-FDDC approach.

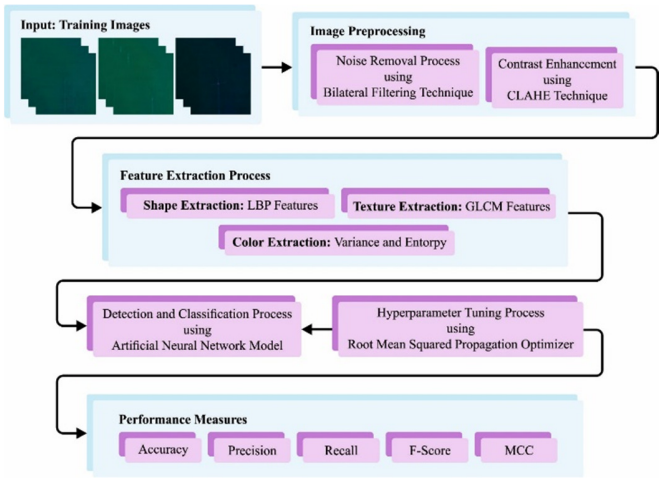

Fig. 1. Workflow of the OANN-FDDC approach.

A. Preprocessing

The preprocessing of the input images is conducted in two approaches: BF and CLAHE-based noise removal and contrast augmentation. BF is a nonlinear filtering model that intends to preserve edges while reducing noise in an image. To achieve this, it considers the spatial distance and intensity differences between adjacent pixels. The filter uses a weighted average for the pixels within the local neighborhood, where the weight depends on the spatial and intensity differences. The underlying concept behind BF is that neighboring pixels with related intensity must contribute more to the filtering method, whereas pixels with considerably different intensities must have a lesser impact. BF efficiently smoothens the image while preserving edges by integrating spatial and intensity information. CLAHE is a contrast enhancement method that enhances the image's local contrast by reallocating pixel intensity. It adapts histogram equalization to work on smaller regions of the images, which allows enhancing the localized contrast. The major objective of CLAHE is to improve the details in an image while avoiding artifacts and excessive noise amplification. It can be accomplished by limiting the maximum amplification possible for all the regions, thus preventing over-amplification of the noise.

B. Feature Extraction

At this stage, three sets of features, namely shape, color, and texture are extracted from the pre-processed fabric images.

1) LBP Features

LBP features contain 2 bitwise transitions, in [0-1] and [1-0]. LBP evaluates the variance and means of the entire intensity of the pixel employing a greyscale imagery as input [21]. The mathematical formula of LBP is:

$$Texture\ Features_{LBP}(\mathfrak{T},\mathcal{R}) = \sum_{\mathfrak{T}=0}^{\mathfrak{T}-1} S\,(\mathfrak{U}_{\mathfrak{T}} - \mathfrak{U}_{\mathfrak{C}})2^{\mathfrak{T}} \quad (1)$$

In (1), $\mathfrak{U}_{\mathfrak{C}}$ shows the intensity contrast, $\mathfrak{T}$ denotes the neighborhood intensity count, $\mathcal{R}$ represents the radius, and $\mathfrak{U}_{\mathfrak{T}}$ indicates the variance of adjacent pixel intensity derived from $(\mathfrak{T},\mathcal{R})$.

$$S_n(\mathfrak{T}) = \begin{cases} 1, \text{if } \mathfrak{T} \geq t \\ 0, \text{otherwise} \end{cases} \quad (2)$$

In (2), the central pixel t is related to the neighboring pixels $S_n(\mathfrak{T})$. It produces a 1×59 feature vector for a single image and N×59 for N images.

2) GLCM Features

GLCM is employed for the extraction of the texture features assuming the 2nd-order relationships among neighboring and reference pixels. GLCM devises the co-occurrence matrix by relating the adjacent pixel values. The row count and column count of matrices are equivalent to the gray level counts [22]. Calculating GLCM for the k^{th} channel of data, G_k, with L levels, 4 features are taken out, correlation (r_k), contrast (c_k), homogeneity (h), and energy (e_k). c_k and r_k are calculated by:

$$c_k = \sum_{i=1}^{L}\sum_{i=1}^{L}(i-j)^2 G_k(i-j)$$

$$r_k = \sum_{i=1}^{L}\sum_{i=1}^{L} G_k\,(i-j)\left(\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^2\sigma_j^2}}\right) \quad (3)$$

where:

$$\mu_i = \sum_{j=1}^{L}\sum_{i=1}^{L} i\,G_k(i,j)$$

$$\mu_j = \sum_{j=1}^{L}\sum_{i=1}^{L} j\,G_k(i,j)$$

$$\sigma_i^2 = \sum_{j=1}^{L}\sum_{i=1}^{L}(i-\mu_i)^2 G_k(i,j)$$

$$\sigma_j^2 = \sum_{j=1}^{L}\sum_{i=1}^{L}(j-\mu_j)^2 G_k(i,j) \quad (4)$$

where the 4×1 vector $v_k = [c_k, r_k, e_k, h_k]^T$ denotes the feature vector for all the channels. Therefore, the feature vector of epochs considering every channel with 60 features is attained using:

$$f_{60\times 1} = [v_1^T, \ldots, v_{15}^T]^T \quad (5)$$

3) Variance and Entropy Features

Color features including variance and entropy are widely applied to calculate the complexity and distribution of colors in an image. Variance measures the variability or spread of color values in an image. It indicates how much the colors deviate from the mean or average color. In terms of color images, variance is separately measured for combinations of channels or all the color channels (i.e. red, green, and blue). High variance values imply a large range of color values, which indicates potentially more diverse vibrant colors in the image. At the same time, a low variance value suggests a more limited range of colors, possibly leading to a subdued or more monochromatic appearance. Entropy is a measure of the quantity of uncertainty or information present in a random variable. In terms of color images, color entropy measures the randomness or complexity of the color distribution. It shows the uniformity and diversity of colors in an image. Entropy is frequently used in computer vision and image analysis tasks to describe the texture, uniqueness, or complexity of an image.

C. Optimal ANN-based Classification

To detect the FDs, an ANN model is used [23]. Generally, the ANN mechanism comprises 3 types of layers, i.e. Hidden Layers (HLs), input layer, and output layer. The hidden and input layers have an unobservable and predictor node and employ a non-linear conversion to the input layer's linear group. The resultant layer has the effect that is any function of a hidden unit. In output and hidden layers, the accurate procedure of function relies on the user definition and network type.

The activation transfer function (g_h) is a sigmoid function from HLs and the output layer (g_0). The i^{th} response y_i for the predictor value H_{ih} is a non-linear function as:

$$y_i = g_0(H_i'\beta) + \varepsilon_i$$

$$H_{ih} = g_h(X_i'\alpha_h) \quad (6)$$

where β indicates the weighted vector of hidden to output units, X_i' denotes the i^{th} row of the input data matrix X, α indicates the weighted matrix of input to hidden units, and H_{ih} is a non-linear function of linear integration of the input dataset:

$$y_i = g_0(H_i'\beta) + \varepsilon_i =$$

$$g_0[\beta_0 + \sum_{h=1}^{H-1}\beta_h\, g_h(X_i'\alpha_h)] + \varepsilon_i \quad (7)$$

Using the sigmoid function, (7) is expressed as:

$$y_i = [1 + \exp(-H_i'\beta)]^{-1} + \varepsilon_i$$

$$= [1 + \exp(-\beta_0 - \sum_{h=1}^{H-1}\beta_h\,[1 + \exp(-X_i'\alpha_j)]^{-1})]^{-1} + \varepsilon_i$$

$$= g(X_i, \beta, \alpha_1, \alpha_2, \ldots, \alpha_H) + \varepsilon_i \quad (8)$$

In (6), $\beta, \alpha_1, \alpha_2, \ldots, \alpha_H$ are unknown parameter vectors, X_i denotes the vector of identified constants, and ε_i represents the residuals. The weights (parameters) are evaluated by enhancing some criterion functions like minimising the sum of squared errors or maximizing the log-likelihood function.

In such problems, overfitting is one of the most serious problems. To overcome it, a penalty term was considered as an optimizer condition. The penalized least square condition is used for estimating parameters:

$$E^* = \sum_{i=1}^{n}(y_i - \hat{y})^2 + p_\lambda(\beta, \alpha_1, \alpha_2, \ldots, \alpha_H)$$

$$= \sum_{i=1}^{n}(y_i - g(X_i, \beta, \alpha_1, \alpha_2, \ldots, \alpha_H))^2$$

$$+ p_\lambda(\beta, \alpha_1, \alpha_2, \ldots, \alpha_H) \quad (9)$$

where the penalty term is: $p_\lambda(\beta, \alpha_1, \alpha_2, \ldots, \alpha_H) = \lambda(\sum \beta_i^2 + \sum \alpha_{ij}^2)$.

Finally, the RMSProp optimizer is employed for the optimal selection of the parameters related to the ANN model [24]. It utilizes a decaying average of partial gradient from the step size adaptation for all the parameters. In the RMSprop, every upgrade was completed based on (10). Upgrading was completed individually for all parameters.

$$v_t = \delta * v_{t-1} - (1-\delta) * g_t^2$$

$$\triangle \omega_t = -\frac{\eta}{\sqrt{y_t+\epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \triangle \omega_t \quad (10)$$

where g_t shows the gradient at t time along w_j, η denotes the rate of learning, and v_t indicates the gradient square's exponential average. In the training process, the weight in the network can be upgraded by shifting the value it can save for all the neurons every time a novel batch of trained data is obtained.

IV. RESULTS AND DISCUSSION

In this section, the simulation outcome of the OANN-FDDC method is investigated by implementing the FD dataset from [25] and the ZJU-Leaper dataset [26]. The features involved are 640 LBP features, GLCM Features (Energy, Contrast, Homogeneity, Correlation, Dissimilarity, Active Shape Models), and Texture Features (Variance, Entropy). Figure 2 depicts sample images of the two databases.

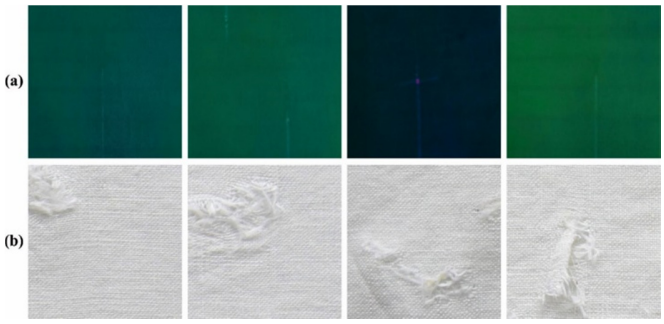

Fig. 2. Sample images: (a) [25], (b) [26].

The suggested technique is put under simulation by employing Python 3.6.5 tool on PC i5-8600k, 250GB SSD, GeForce 1050Ti 4GB, 16GB RAM, and 1TB HDD. The parameter set up is: learning rate: 0.01, activation: ReLU, epoch count: 50, dropout: 0.5, and size of batch: 5.

In Table I, the complete outputs of the OANN-FDDC technique on the dataset from [25] are shown. The outputs indicate that the OANN-FDDC technique accomplishes enhanced performance. On TRP, the OANN-FDDC technique achieves an $accu_y$ of 96.97%, $prec_n$ of 99.02%, $reca_l$ of 96.97%, F_{score} of 97.92%, and MCC of 97.19%. Also, on TRP, the OANN-FDDC method accomplishes $accu_y$ of 96.97%, $prec_n$ of 99.02%, $reca_l$ of 96.97%, F_{score} of 97.92%, and MCC of 97.19%.

Table II shows the results of the total analysis of the OANN-FDDC method on ZJU-Leaper dataset [26]. The resultant represented the OANN-FDDC method to achieve improved performance. On TRP, the OANN-FDDC method obtains $accu_y$ of 94.78%, $prec_n$ of 95.17%, $reca_l$ of 94.78%, F_{score} of 94.72%, and MCC of 89.96%. Moreover, on TRP, the OANN-FDDC method obtains $accu_y$ of 95.46%, $prec_n$ of 96.02%, $reca_l$ of 95.46%, F_{score} of 95.55%, and MCC of 91.48%.

In Table III, the relative outputs of the OANN-FDDC models are compared with those of recent approaches [27, 28]. The simulation outcome indicates that the CNN, ResNet50v2, and DenseNet169v2 models gave poor performance, whereasFPN, Bi-FPN, NAS-FPN, and Dense-FPN models reported moderately greater outputs. The i-FPN model accomplishes considerable performance with an $accu_y$ of 94.20%, $prec_n$ of 89.16%, $reca_l$ of 90.05%, and $F1_{score}$ of 92.34%, but the OANN-FDDC method reaches maximum results with $accu_y$ of 96.97%, $prec_n$ of 99.02%, $reca_l$ of 96.97%, and $F1_{score}$ of 97.92%. These results illustrate the supreme accomplishment of the OANN-FDDC method in terms of various performance metrics.

TABLE I. CLASSIFICATION OUTPUT OF OANN-FDDC METHODOLOGY ON [25]

Metric	Training Phase	Testing Phase
Accuracy	96.97	95.83
Precision	99.02	97.62
Recall	96.97	95.83
F-Score	97.92	96.54
MCC	97.19	94.99

TABLE II. CLASSIFICATION OUTPUT OF OANN-FDDC METHODOLOGY ON [26]

Metrics	Training phase	Testing phase
Accuracy	94.78	95.46
Precision	95.17	96.02
Recall	94.78	95.46
F-Score	94.72	95.55
MCC	89.96	91.48

TABLE III. RELATIVE OUTPUT OF OANN-FDDC METHODOLOGY WITH CURRENT MODELS

Techniques	$Accu_y$	$Prec_n$	$Reca_l$	$F1_{score}$
CNN	67.00	70.00	66.00	79.00
RestNet50v2	56.00	56.00	55.00	55.00
DenseNet169v2	56.00	59.00	55.00	57.00
FPN	83.40	90.44	92.43	89.72
Bi-FPN	85.20	89.95	89.75	89.71
NAS-FPN	87.90	89.03	90.26	91.49
Dense-FPN	90.20	91.92	91.95	91.38
i-FPN	94.20	89.16	90.05	92.34
OANN-FDDC	96.97	99.02	96.97	97.92

V. CONCLUSION

This study introduces the novel OANN-FDDC methodology for automated FD detection and classification. The OANN-FDDC methodology utilizes handcrafted features with a hyperparameter-tuning procedure for effectually detecting FDs. To manage this, the OANN-FDDC technique comprises pre-processing, handcrafted extraction, and ANN

and RMSProp classification and tuning. In order to enhance the recognition outputs of the ANN method, the RMSProp optimizer was used for the parameter selection process. The experimental results of the OANN-FDDC technique produced an outcome of 96.97%, which surpasses other known approaches.

The OANN-FDDC model excels in FD detection but may struggle with intricate patterns and lighting variations, necessitating enhanced robustness for diverse operational settings. In the future, DL methods will improve the performance of the OANN-FDDC method and adaptive learning mechanisms for handling diverse fabric patterns will be included and its robustness in dynamic lighting conditions will be refined.

REFERENCES

[1] Y. Kahraman and A. Durmusoglu, "Deep learning-based fabric defect detection: A review," *Textile Research Journal*, vol. 93, no. 5–6, pp. 1485–1503, Mar. 2023, https://doi.org/10.1177/00405175221130773.

[2] M. Chen *et al.*, "Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization," *Computers in Industry*, vol. 134, Jan. 2022, Art. no. 103551, https://doi.org/10.1016/j.compind.2021.103551.

[3] L. Cheng, J. Yi, A. Chen, and Y. Zhang, "Fabric defect detection based on separate convolutional UNet," *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 3101–3122, Jan. 2023, https://doi.org/10.1007/s11042-022-13568-7.

[4] Q. Liu, C. Wang, Y. Li, M. Gao, and J. Li, "A Fabric Defect Detection Method Based on Deep Learning," *IEEE Access*, vol. 10, pp. 4284–4296, 2022, https://doi.org/10.1109/ACCESS.2021.3140118.

[5] T. Meeradevi, S. Sasikala, S. Gomathi, and K. Prabakaran, "An analytical survey of textile fabric defect and shade variation detection system using image processing," *Multimedia Tools and Applications*, vol. 82, no. 4, pp. 6167–6196, Feb. 2023, https://doi.org/10.1007/s11042-022-13575-8.

[6] J. Jing, Z. Wang, M. Rätsch, and H. Zhang, "Mobile-Unet: An efficient convolutional neural network for fabric defect detection," *Textile Research Journal*, vol. 92, no. 1–2, pp. 30–42, Jan. 2022, https://doi.org/10.1177/0040517520928604.

[7] H. Xie and W. Xu, "Effective fabric defect detection using contrastive learning and layered fusion network," in *Fourteenth International Conference on Graphics and Image Processing*, Nanjing, China, Oct. 2022, vol. 12705, pp. 199–208, https://doi.org/10.1117/12.2680579.

[8] K. Pandya, D. Dabhi, P. Mochi, and V. Rajput, "Levy Enhanced Cross Entropy-based Optimized Training of Feedforward Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 12, no. 5, pp. 9196–9202, Oct. 2022, https://doi.org/10.48084/etasr.5190.

[9] G. Anuradha and D. N. Jamal, "Classification of Dementia in EEG with a Two-Layered Feed Forward Artificial Neural Network," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7135–7139, Jun. 2021, https://doi.org/10.48084/etasr.4112.

[10] Y. Ivanova, "Damage Detection in Free–Free Glass Fiber Fabric Composite Beams by measuring Flexural and Longitudinal Vibrations," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10685–10690, Jun. 2023, https://doi.org/10.48084/etasr.5788.

[11] 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, Aug. 2021, Art. no. 114838, https://doi.org/10.1016/j.eswa.2021.114838.

[12] N. Alruwais *et al.*, "Hybrid mutation moth flame optimization with deep learning-based smart fabric defect detection," *Computers and Electrical Engineering*, vol. 108, May 2023, Art. no. 108706, https://doi.org/10.1016/j.compeleceng.2023.108706.

[13] B. Fang, X. Long, F. Sun, H. Liu, S. Zhang, and C. Fang, "Tactile-Based Fabric Defect Detection Using Convolutional Neural Network With Attention Mechanism," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–9, 2022, https://doi.org/10.1109/TIM.2022.3165254.

[14] X. Jun, J. Wang, J. Zhou, S. Meng, R. Pan, and W. Gao, "Fabric defect detection based on a deep convolutional neural network using a two-stage strategy," *Textile Research Journal*, vol. 91, no. 1–2, pp. 130–142, Jan. 2021, https://doi.org/10.1177/0040517520935984.

[15] Z. Wang, J. Junfeng, H. Zhang, and Y. Zhao, "Real-Time Fabric Defect Segmentation Based on Convolutional Neural Network," *AATCC Journal of Research*, vol. 8, no. 1_suppl, pp. 91–96, Sep. 2021, https://doi.org/10.14504/ajr.8.S1.12.

[16] D. He, J. Wen, and Z. Lai, "Textile Fabric Defect Detection Based on Improved Faster R-CNN," *AATCC Journal of Research*, vol. 8, no. 1_suppl, pp. 82–90, Sep. 2021, https://doi.org/10.14504/ajr.8.S1.11.

[17] H. Zhang, S. Wang, S. Lu, L. Yao, and Y. Hu, "Attention-Gate-based U-shaped Reconstruction Network (AGUR-Net) for color-patterned fabric defect detection," *Textile Research Journal*, vol. 93, no. 15–16, pp. 3459–3477, Aug. 2023, https://doi.org/10.1177/00405175221149450.

[18] H. Zhang, G. Qiao, S. Lu, L. Yao, and X. Chen, "Attention-based Feature Fusion Generative Adversarial Network for yarn-dyed fabric defect detection," *Textile Research Journal*, vol. 93, no. 5–6, pp. 1178–1195, Mar. 2023, https://doi.org/10.1177/00405175221129654.

[19] A. Suryarasmi, C.-C. Chang, R. Akhmalia, M. Marshallia, W.-J. Wang, and D. Liang, "FN-Net: A lightweight CNN-based architecture for fabric defect detection with adaptive threshold-based class determination," *Displays*, vol. 73, Jul. 2022, Art. no. 102241, https://doi.org/10.1016/j.displa.2022.102241.

[20] C. C. Ho, W. C. Chou, and E. Su, "Deep convolutional neural network optimization for defect detection in fabric inspection," *Sensors*, vol. 21, no. 21, Nov. 2021, Art. no. 7074, https://doi.org/10.3390/s21217074.

[21] U. Ishtiaq, E. R. M. F. Abdullah, and Z. Ishtiaque, "A Hybrid Technique for Diabetic Retinopathy Detection Based on Ensemble-Optimized CNN and Texture Features," *Diagnostics*, vol. 13, no. 10, Art. no. 1816, May 2023, https://doi.org/10.3390/diagnostics13101816.

[22] S. M. Mousavi, A. Asgharzadeh-Bonab, and R. Ranjbarzadeh, "Time-Frequency Analysis of EEG Signals and GLCM Features for Depth of Anesthesia Monitoring," *Computational Intelligence and Neuroscience*, vol. 2021, Aug. 2021, Art. no. e8430565, https://doi.org/10.1155/2021/8430565.

[23] A. Biglarian, E. Bakhshi, A. R. Baghestani, M. R. Gohari, M. Rahgozar, and M. Karimloo, "Nonlinear Survival Regression Using Artificial Neural Network," *Journal of Probability and Statistics*, vol. 2013, Feb. 2013, Art. no. e753930, https://doi.org/10.1155/2013/753930.

[24] F. Mehmood, S. Ahmad, and T. K. Whangbo, "An Efficient Optimization Technique for Training Deep Neural Networks," *Mathematics*, vol. 11, no. 6, Jan. 2023, Art. no. 1360, https://doi.org/10.3390/math11061360.

[25] S. Ranathunga, "Fabric Defect Dataset." [Online]. Available: https://www.kaggle.com/datasets/rmshashi/fabric-defect-dataset.

[26] C. Zhang, S. Feng, X. Wang, and Y. Wang, "ZJU-Leaper: A Benchmark Dataset for Fabric Defect Detection and a Comparative Study." Dec. 2020, [Online]. Available: http://www.qaas.zju.edu.cn/zju-leaper/.

[27] J. Xiang, R. Pan, and W. Gao, "Online Detection of Fabric Defects Based on Improved CenterNet with Deformable Convolution," *Sensors*, vol. 22, no. 13, Jun. 2022, Art. no. 4718, https://doi.org/10.3390/s22134718.

[28] A. Amelio *et al.*, "Defining a deep neural network ensemble for identifying fabric colors," *Applied Soft Computing*, vol. 130, Nov. 2022, Art. no. 109687, https://doi.org/10.1016/j.asoc.2022.109687.