Optimized Attention Augmented Residual Convolutional Neural Network with FA-ResNet for Fabric Defect Detection

Sujitha Ravichandran * **Venkatasalam Kandasamy** **

* *Assistant Professor, Department Of Computer Science and Engineering, Annapoorana Engineering College, Periya Seeragapadi, Salem, Tamil Nadu, India. (rsujitha2025@gmail.com).*
** *Professor, Department Of Computer Science and Engineering, Mahendra Engineering College, Namakkal, Tamil Nadu, India. (venkispkm@gmail.com)*

Abstract: Fabric defects have a significant effect on a textile product's quality. Manufacturers identify inefficiencies in their processes, equipment failures, or mistakes made by operators by analyzing the types and quantities of defects. While modern algorithms offer high accuracy in classifying fabric surface anomalies, certain defects remain challenging to detect due to their complex patterns, subtle features, or visual obstructions. This study explores advanced detection methods aimed at improving the reliability and precision of automated fabric inspection systems. In this manuscript, Optimized Attention Augmented Residual Convolutional Neural Network with FA-ResNet for Fabric Defect Detection (AARCNN-FA-ResNet-FDD) is proposed. Initially, Input images are collected using Fabric Defect Dataset. Input images are pre-processed using Multiple Local particle Filter (MLPF) to eliminate the unnecessary noise from the images. The pre-processing images are supplied into feature extraction. Then, feature extraction is done by Synchro Transient Extracting Transform (STET) to extract texture features like Entropy, Correlation, Contrast, Homogeneity. The extracted features are supplied to Attention Augmented Residual Convolutional Neural Network using Feature Affine Residual Network (AARCNN-FA-ResNet) for detecting the fabric surface as captured, hole, horizontal and vertical. In general, AARCNN-FA-ResNet does not express some adaption of optimization techniques to detect the ideal parameters to ensure accurate fabric surface classification. Lotus effect optimization algorithm (LEOA) is used for optimizing the weight parameter of AARCNN-FA-ResNet classifier, which precisely classifies the fabric surface. The proposed method achieves an efficiency improvement of 99% in fabric surface classification accuracy compared to the existing methods. The proposed AARCNN-FA-ResNet-FDD method is examined using performance metrics like accuracy, precision, sensitivity, specificity, f1-score, computation time and ROC. Performance of the AARCNN-FA-ResNet-FDD approach attains16.43%, 24.29% and 30.82% higher accuracy; 18.64%, 24.88% and 31.95% higher precision and 32.22%, 32.60% and 27.05% lower computation time compared with existing methods like FSDC-CSO-DRN (Fabric Surface Defect Categorization and systematic analysis utilizing Cuckoo Search Optimized Deep Residual Network) , LSTM-TC-FDD (LSTM dependent Texture Categorization and Fabric Defect Detection), CNN-ATCD-FDD (lightweight CNN-based method for Fabric Defect Detection along with Adaptive Threshold-based Class Determination) respectively.

Keywords: Fabric defect detection, AARCNN-FA-ResNet-FDD, Lotus Effect Optimization Algorithm, Multiple Local Particle Filter, Convolutional neural network, Texture analysis

1. INTRODUCTION

The textile industry produces goods on a vast scale using intricate procedures but it is heavily affected by the defects in the fabric which reduce the product values and raw material wastage. A fabric defect is a defect on the outside of a manufactured material Fang et al. (2022). Fabric faults are common, and the majority of them are the product of production or equipment malfunctions Chen et al. (2022). Flaws can also be caused by machine spoilage or faulty yarns Pourkaramdel et al. (2022). Every element has a distinct effect and considerably lowers the fabric's usability and sales. To overcome these kind of challenges it is necessary to identify the defects at its earliest stage Althubiti et al. (2022). The qualities of a textile product are considerably impacted by faults in the fabric. Waste and higher production costs are two outcomes of defective fabrics Singh and Desai (2023). The later step of the fabric production process is harmed by the errors in the earlier stage. Classifying fabric defects is essential for preserving product quality, cutting expenses, guaranteeing consumer safety Li et al. (2023) enhancing driving procedures, fulfilling legal obligations, and promoting efficient supplier management Yu et al. (2022)Wang et al. (2022)Mahmood et al. (2023). By precisely categorizing flaws and implementing the necessary remedial measures, manufacturers can pinpoint the underlying causes Lu et al. (2023). This results in lower material waste, more effective production and financial savings. By spotting the flaws, one can make sure that only premium fabrics are used to produce the goods. Early detection of fabrication flaws could there by lessen the losses incurred by companies Jha and Babiceanu (2023). With the complexities involved in textile defect identification, emerging deep learning technologies are proving to be valuable tools for improving early detection accuracy. Deep learning has significantly outperformed predictions with respect to industrial automation Üzen et al. (2023). Deep learning uses signals and visuals to improve manufacturing and lessen industry challenges Gao et al. (2023). Historically, the human process of identifying fabric flaws involved costly and inefficient eye inspection. When an issue is identified, the production process is halted, and defects location and circumstances are recorded Meeradevi and Sasikala (2024). Manual processing possess drawbacks and production efficiency is lowered Zhao et al. (2024). Smaller flaws are unnoticed by the operator, or it might take a lot of work to pinpoint the fabric flaw. At the moment, fabric defect classification and segmentation depend heavily on deep learning. Deep learning networks frequently use an amount of hyper-parameters, like learning rate, regularization parameter, network depth, amount of filters, filter size Liu et al. (2022). These characteristics are specified in relation to general terms and are usually acquired through experience, have a major impact on the classification's efficacy. Determining the ideal value for these hyper parameters is a big and difficult challenge Shao et al. (2022). These hyper parameters are divided into three categories: integer, continuous and mixed depending on the value types assigned to them. The main application of the gradient algorithm is in continuous hyper parameter optimization Fouda (2022). For hyper parameter optimization, the majority of studies employ genetic and evolutionary algorithms based on particle swarming Zhao et al. (2023). This research employs the new technique for hyper parameter optimization since the algorithm has shown to perform better than existing intelligent evolutionary algorithms to be useful in resolving a variety of optimization problems Liu and Li (2022).

1.1 Problem statement and Motivation behind this Research work

In the textile industry, limited data size makes the network less generalizable further raises the risk of over fitting. The network's upper layers are retained and retrained to keep course characteristics related to defect features that cause feature detection to lag. A data imbalance issue arise if there is a significant disparity between the entire amount of images in the dataset and their sizes. Throughout the data collection procedure, the fabric skews in both directions. The random auto-orientation also plays a role in the orientation variation and position of the does not defect throughout surface, in addition to flipping in both directions. It addresses the network overfitting issue. The study tracked the accuracy and loss of training and validation for every epoch and created a robust solution against the previously mentioned ill-posed region in order to prevent overfitting problems. This paper proposed AARCNN-FA-ResNet-FDD. Here, a novel method utilizing an AARCNN-FA-ResNet is presented. Texture patterns have the ability to display minute details that are essential for precise classification. FA-ResNet's AARCNN architecture makes it excellent at gathering precise data. FA-ResNet learns hierarchical representations of texture patterns by stacking numerous residual blocks, progressively capturing both lower and higher level characteristics. It makes it easier for the AARCNN to comprehend and distinguish between distinct texture patterns, including ones that are variants.

1.2 Contribution

- This manuscript proposes the AARCNN-FA-ResNet-FDD method.
- STET to extract the term texture features, it penalizes text differences that belong on the same scene plane but don't follow the same model's parameters.
- The initial point of novelty and significance by AARCNN-FA-ResNe is the concept of combining as it develops in the framework that is demonstrably capable of detecting the Fabric Defecton fabric surface.
- The proposed AARCNN-FA-ResNet model is a significant contribution in and of itself as it can outperform another model with a similar structure and requires fewer cumbersome AARCNN-FA-ResNet layers.

The remaining sections are structured as: part2 defines literature review, part 3 describes the proposed approach, part 4 proves outcomes with discussions, part 5 presents conclusion.

2. LITERATURE REVIEW

Numerous studies efforts in the literature indicated Fabric defect detection using deep learning; here, a few recent studies are revised below. In 2024, Mewada, et.al, Mewada et al. (2024) have presented FSDC-CSO-DRN. A CNN technique depending on residual networks to improve fabric defect categorization was presented Four types of fabric defects are identified by fine-tuning a pretrained residual network, ResNet50: holes, objects, oily stains, and stitching mistakes on the cloth's exterior. Using cuckoo search optimization, which employs categorization error as fitness function, the fine-tuned network was further optimized. It attains higher accuracy and it attains lower precision. In 2023, Kumar, et.al, Kumar and Bai (2023) have presented LSTM based texture categorization and error identification in a fabric. This publication use sub band decomposition and multi-scale curvelet image decomposition to detect fabric flaws and patterns. There are two stages involved in fault identification. The analysis of flawless images was the initial step. In the second step, logistic regression was used to identify the flaws through the use of LSTM techniques. The suggested approach identifies defects and recognizes patterns in a shorter amount of computational time by breaking down the trained images into blocks. It attains higher precision and it attains lower specificity. In 2022, Suryarasmi, et.al, Suryarasmi et al. (2022) have presented FN-Net: CNN-ATCD-FD:

lightweight CNN-based architecture to identify fabric errors. The suggested FN-Net complete training three to thirty-three times faster than state-of-the-art structures like VGG16, MobileNetV2, Efficient Net, Dense Net while consuming less RAM and graphics processing unit. It attains high sensitivity and attains lower F1-Score. In 2022, YaşarÇıklaçandır, et.al, YaşarÇıklaçandır et al. (2024) have presented determination of frequent fabric fault sutilizing different machine learning technique. Here, it employs a variety of feature sets, including those derived from Deep Learning Architectures, Principal Component Analysis, Grey Level Co-occurrence Matrix, Discrete Cosine Transform, to identify errors. Three distinct classification methods (K-Nearest Neighbor, Support Vector Machine, Decision Tree) have been used to categorize the retrieved characteristics. It aims to explore the implications of numerous approaches for feature extraction. An analysis of the various features extraction techniques and classifiers' performances was presented. It attains lower computation time and lower ROC. In 2022, Seçkin, et.al, ¸C. Seçkin and Seckin (2022) have presented fabric defect detection using entwined frame vector feature extraction. In this case, the features are vector functions between the frame vectors. The developed method was essentially an image window feature extraction process. The matrix defining the window contains entwined frames. Algorithms for categorization employ these features to find defects. Performance and time comparisons between the suggested method and conventional texture feature extraction techniques were conducted using the AITEX dataset. It attains higher ROC and it attains higher computation time. In 2022, Jain, et.al, Jain et al. (2022) have presented deep learning-dependent artificial data augmentation for surface fault identification and categorization. The Generative Adversarial Networks (GANs) was used in data augmentation to produce artificial images. For hot-rolled steel strip data from NEU Surface Defect Database, Northeastern University (China) Categorization (NEU-CLS) dataset and the entire process of data augmentation was performed after training three GAN architectures. A basic CNN architecture classification accuracy was assessed using artificially enhanced data and contrasted with related state-of-the-arts. The presented GANs-based augmentation approach considerably raises CNN's categorization of surface fault performance. It achieves greater specificity and lesser accuracy. In 2022, Cheng, L., et.al, Cheng et al. (2023) have presented fabric defect detection depending on separate convolutional UNet. The defect detection was a crucial but challenging task in the textile industry. Manual examination was the basis of traditional methods expensive also harmful to the fabric. Semantic segmentation network-based deep learning methods were used to simply and effectively implement the high accuracy fabric defect identification. A Separation Convolution UNet (SCUNet) with a mere 4.27 M (million) parameters was paired with depth-separable convolution, convolutional down sampling, and a cross-parallel ratio loss function (IoU Loss). By extracting surface elements from fabric images, fabric flaw locations can be found. It achieves higher F1-Score and lower sensitivity.

Table 1 presents a literature review of various approaches for fabric defect detection. Mewada et al. [21] optimize a deep residual network with cuckoo search optimization, achieving higher accuracy but lower precision. Kumar et al. [22] use LSTM for fabric texture classification, detecting defects with sub-band decomposition, offering higher precision but lower specificity. Suryarasmi et al. [23] develop FN-Net, a lightweight architecture that outperforms VGG16 and MobileNetV2 in speed and resource efficiency, though it has a low F1-score. YaşarÇıklaçandır et al. [24] compare machine learning approaches like KNN, SVM, and Decision Tree, improving defect identification with lower computation time but lower ROC. Seçkin et al. [25] enhance defect detection using window-based feature extraction, improving detection over traditional techniques but with higher computation time. Jain et al. [26] augment synthetic data with GANs for surface defect detection, improving classification accuracy but sacrificing accuracy. Cheng et al. [27] implement SCUNet for fabric defect detection, enabling efficient detection with higher F1-score but lower sensitivity

3. PROPOSED METHODOLOGY

In this section, the AARCNN-FA-ResNet-FDD is proposed. There are five steps in this process: data Acquisition, pre-Processing, feature extraction, fabric defect detection, optimization. The proposed method is used to analyze the fabric defect detection undergo pre-processing for further analysis. Then the features are extracted using STET. The final step involves employing a AARCNN-FA-ResNet to detect the fabric defect detection as captured, hole, horizontal and vertical. The LEOA method is used to improve the AARCNN-FA-ResNet. Figure 1 depicts block diagram of proposed AARCNN-FA-ResNet-FDD approach. The detailed description of each step is provided below,

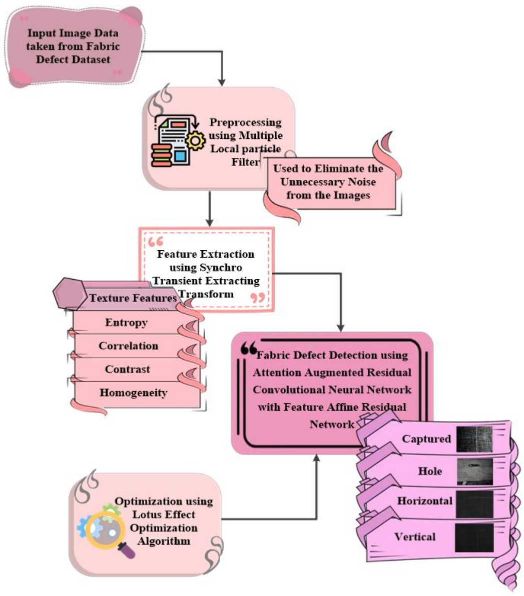

Fig. 1. Block Diagram for Proposed AARCNN-FA-ResNet-FDD Method

3.1 Image Acquisition

The input image is collected via fabric defect dataset Kaggle User: rmshashi (2022). The dataset is provided

Table 1. Literature review table

Author	Objectives	Methods	Key Finding	Advantage and Disadvantages
Mewada, et.al.	Optimize deep residual network for fabric defect classification.	Deep Residual Network with cuckoo search optimization.	ResNet50 fine-tuned with cuckoo search optimizes fabric defect classification.	**Advantage:** It achieves greater accuracy **Disadvantage:** It achieves lesser precision.
Kumar, et.al.	Utilizes LSTM for fabric texture classification and detection.	Long Short Term Memory (LSTM).	Sub-band decomposition and LSTM with logistic regression detect fabric defects.	**Advantage:** It achieves greater precision. **Disadvantage:** It achieves lesser specificity.
Suryavanshi, et al.	To develop a lightweight FN-Net for adaptive fabric defect detection.	Lightweight Convolutional Neural combined with FN-Net.	FN-Net outperforms VGG16, MobileNetV2 in speed and resource efficiency.	**Advantage:** High sensitivity. **Disadvantage:** Low F1-Score.
Yaşarçukacandır, et al.	To compare machine learning techniques for fabric defect identification.	K-Nearest Neighbor, Support Vector Machine, Decision Tree.	Multiple feature extraction techniques and classifiers improve fabric defect identification.	**Advantage:** It attains lower computation time. **Disadvantage:** It attains lower ROC.
Seçkin, et al.	Extract intertwined frame vector features for fabric defect detection.	K Nearest Neighbors (KNN) and Random Forest (RF) algorithms.	Window-based feature extraction improves defect detection compared to traditional techniques.	**Advantage:** It attains higher ROC. **Disadvantage:** It attains higher computation time.
Jain, et al.	Augment synthetic data for surface defect detection and classification.	Convolutional Neural Networks (CNN) and GANs.	GAN-based augmentation improves CNN classification accuracy for surface fault detection.	**Advantage:** It achieves higher specificity. **Disadvantage:** It achieves lower accuracy.
Cheng, L., et al.	To implement separate convolutional UNet for fabric defect detection.	SCUNet with convolutional down sampling and depth-separable convolution.	SCUNet enables efficient, accurate fabric defect detection using semantic segmentation.	**Advantage:** It attains higher F1-Score. **Disadvantage:** It attains lower sensitivity.

using the computer science and engineering department at the University of Moratuwa's Intelligence Lab. For every defective picture sample, there are three mask images and this fabric defect databases three classes of faulty images horizontal, vertical, and holes. The raw photos are in the "captured" folder and the dimensions of images are 640x360. Here, Validation images (30%) and 70% of the training images from the dataset were used in the experiment.

3.2 Pre-Processing using Multiple Local Particles Filter

The MLPF Li et al. (2024) technique utilized to eliminate the unnecessary noise from the collected input image is discussed. To increase image quality and improve flaw identification accuracy, fabric defect detection employing pre-processing with a required number of meticulous processes. First, high-resolution image of the cloth are taken, which could contain distortions and noise. To begin the pre-processing stage, scattering must be initialized, a collection of particles throughout the images using equation (1).

$$S(A_t|B_{1:t-1}) = \int S(A_t|A_{t-1})S(A_{t-1}|B_{1:t-1})\, dA_{t-1} \quad (1)$$

Where, indicates the total amount of input image collected; indicates the dynamically updated with each repetition's of image data; indicates the local bandwidth for channel between pixel and neighbor pixel and indicates the dimension of each pixel. Every particle is an estimate of a potential pixel value. Particles are generated for each pixel or local region, and their weights are assigned according to fabric images likely they are to match the true values of the pixels. Noise and other visual distortions are considered in this weighting using equation (2).

$$(A_i|B_{i,j}) \approx \frac{1}{n}\sum_{i=1}^{n} \delta(A_i - A_i^j) \quad (2)$$

Where, A_i^j indicates the i^{th} and j^{th} pixel depth size of each fabric image; nindicates the n^{th}image pixel from collected input and δ indicates the parameter of MLPF. The scalar fading factor is generated using the adjusted measurement of each unwanted noise removal of images and the covariance in pixel matrix generated by the noise exponential weighting technique using equation (3).

$$u_i^j \alpha = \frac{S(A_i^j|B_{i,j})}{Q(A_i^j|B_{i,j})} \quad (3)$$

Where, u_i^j indicates the normalized form of the input weight image; α indicates the probability function andQ indicates the pixel-wise plane in each image data. Then, to update their estimates and reduce noise, the particles take into account information from nearby pixels. The revised particles are combined to create a pixel value representation that is more accurate and improves the contrast and visibility of any possible flaws. It is possible to apply more noise reduction techniques to further clean the image while keeping significant details intact using equation (4).

$$S(A_{0,i}|B_{1,i}) \approx \sum_{i=1} u_i^j \delta(A_{0:i} - A_{6:i}) \quad (4)$$

Where, u_i^j indicates the unwanted image data obtained from the gathered input data. Unwanted image estimation is the final step in the preparation process because the retained image data is mistakenly identified the unnecessary noise image and cannot be determined by usually based on unwanted noise estimation. The previously described is made worse by the predicted cleaning level's steep fall, and unnecessary noise in image is removed using equation (5).

$$S(A_i|B_{1,i}) = \prod_{j=1}^{N} S(A_i^j|B_{1:i}) \quad (5)$$

Where,$\prod_{j=1}^{N}$ indicates the removal of unnecessary noise from image. By processing MLPF method removes the unnecessary noise from the collected input image. The pre-processing imagery Is given to feature extraction.

3.3 Feature Extraction using Synchro Transient Extracting Transform

This section, STET Ma et al. (2024) is discussed to extract the texture features like entropy, correlation, contrast, homogeneity. A sophisticated method called STET uses the concepts of time-frequency analysis to extract features from images. Because STET operates in both the temporal and frequency domains, unlike traditional approaches which only function in the spatial domain, it is able to capture transient features and fluctuations in the fabric texture more efficiently using equation (6).

$$\hat{X}(R,u) = \begin{cases} \partial_R \hat{u}(T,u)/\partial_R \hat{R}(R,u), if \partial_R \hat{R}(R,u) \neq 0 \\ Inf, if \partial_R \hat{R}(R,u) = 0 \end{cases} \quad (6)$$

Where,$\hat{X}(R,u)$ indicates the processed image data;∂_R represents the phase-aligned linear image data; $\hat{\omega}$ indicates the fabric images that were extracted longitudinally and $\hat{R}$signifies the entire amount of fabric present in the each image pixel. In the textile industry, identifying fabric defects is essential to guaranteeing that the quality of the fabric fulfils the necessary requirements. STET approaches frequently depend on rudimentary image processing techniques or visual inspection, both of which can be unreliable and inefficient. Innovative techniques for extracting features from fabric images, such the TET, provide increased resilience and precision by identifying complex patterns and flaws using equation (7).

$$r_f(R,u) = \begin{cases} U(R,u)\, \partial(u - \hat{u}(R,u)), \&\text{if } \left|\hat{X}(R,u)\right| > \beta^{-2/3} \\ 0, \&\text{Otherwise} \end{cases} \quad (7)$$

Where,r_f indicates the ratio of the pixels from each fabric images; U indicates each feature's threshold value; and $\beta^{-2/3}$ indicates the longitudinal synchro-transient extracting operator. Any image can be computed by splitting it into pixel-by-pixel or neighbor pixel, and then computing the images of the categorical or non-categorical, and using these formulas to generate fabric extract images whose boundaries are then gathered using equation (8).

$$(\hat{R}(R,u)-R)\partial_R\hat{u}(R,u)/\partial_R\hat{R}(T,u) = X^2\beta^2(u-z-XR)(1+y^2\beta^2)^{-1} \tag{8}$$

Where, β^2 indicates the flat surface of the error images and positive images and z indicates the image planar regions in the scene. Although this pixel-by-pixel method is computationally effective, no single image contains all the information required to detect the transmission. Therefore, improper transmission information leads to overexposed features in fabric images. However, each fabric image extraction procedure adjusts the multilevel thresholding objective pixel in order to apply a patch-wise strategy using equation (9).

$$U(R,z+XR) = q_e(R)\sqrt{2\pi\beta(1-i\beta y)^{-1}} \tag{9}$$

Where, $q_\varepsilon(T)$ denotes the text that are rearranged in pixel size in each image and $\sqrt{2\pi\beta(1-i\beta y)^{-1}}$ represents each sentences pixel-by-pixel plane properties at a range of fabric pixel image sizes. The distribution of colors or pixel intensities in an image is measured by entropy. It shows how much information is needed to explain the image; higher entropy denotes greater complexity and lower entropy denotes less complexity using equation (10).

$$Entropy = \sum\sum b(x,y)\log b(x,y) \tag{10}$$

Where,$\log b(x,y)$ denotes range of pixel intensity values that are in fabric images. Correlation evaluates the strength and way of a linear link among two pixels. In the context of picture extraction, these pixel values from different regions inside the same image or between each individual image using equation (11).

$$Correlation = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{m-1}(x-s_x)(y-s_y)b(x,y)}{\sigma_x\sigma_y} \tag{11}$$

Here, s_x and s_y is the positive value means that there is a tendency for the pixel values in one location to increase and vice versa and$\sigma_x\sigma_y$ denotes the pixel intensities' standard deviation. The difference in pixel intensities or colors throughout a picture is brought to light by contrast. Significant contrasts between light and dark areas in an image help to highlight edges and details. Images with low contrast have smoother, less distinct appearances because the transitions between intensities are more gradual using equation (12).

$$Contrast = \sum(x,y)^2 b(x,y) \tag{12}$$

Here,$\sum(x,y)^2$ are the localized regions of the image; contrast can be computed as the difference among the maximum and minimum pixel values in those regions. The degree to which pixel values are dispersed uniformly throughout an area or throughout the image is measured by homogeneity. While low homogeneity denotes greater variance and less uniformity, high homogeneity shows that pixel values are relatively similar, producing a smooth or uniform texture using equation (13).

$$Homogenity = \sum_{x,y}\frac{b(x,y)}{1+|y-x|} \tag{13}$$

Where, $\sum_{x,y}$ is the likelihood that pixel pairings with values x and y will occur at a particular spatial relationship and $|y-x|$ indicates the contribution in each pixel pairs. Then the extracted features are given to AARCNN-FA-ResNet to detect the mentioned Fabric Defect detections.

3.4 Fabric Defect Detection using Attention Augmented Residual Convolutional Neural Network with Feature Affine Residual Network

In this section, AARCNN-FA-ResNet Zhang et al. (2023) is discussed for detecting the fabric surface as captured, hole, horizontal and vertical. To create a AARCNN based fabric defect detection system, begin by gathering the extracted of fabric images, making sure to include captured, hole, horizontal and vertical fabrics. The next important step is to annotate this image with the locations of flaws. Depending on the required level of detection granularity, this can be done using bounding boxes or pixel-by-pixel annotations using equation (14).

$$v_{m.n}^{Q_{g,v}} = \frac{O_{g,v}.L_{m,n}}{\sqrt{N^o}} \tag{14}$$

Where, $v_{m.n}^{Q_{g,v}}$ represents the pixel m,nin the attention weight's width and height parameters; $O_{g,v}$signifies the height, width and channels of the input pixel images; $L_{m,n}$indicates the higher level images in inner layer and $\sqrt{N^o}$is the pixel-by-pixel annotation of each input images. Using batch normalization, residual blocks with convolutional layers, and skip connections, and softmax layer are utilized to mitigate the vanishing gradient issue while capturing higher-level information. Subsequently, attention techniques are incorporated into the residual blocks to concentrate on noteworthy portions of the image and feature layers using equation (15).

$$\tilde{v}^{Q_{g,v}} = \text{Softmax}_{2d}(v^{Q_{g,v}}) \tag{15}$$

Where, $\hat{v}^{Q_{g,v}}$ indicate the difference in the width and height between each pixel; $\text{Softmax}_{2\text{d}}$indicates the soft max layer in AARCNN and $v^{Q_{g,v}}$is the attention block representing the pixel wise images accordingly. By creating attention maps and weighting the feature maps appropriately, spatial attention draws attention to significant regions of the fabric images. Conversely, layer attention uses attention weights that are computed and applied to highlight important feature layers. Following multiple residual and attention-augmented blocks, global average pooling is used for global feature aggregation, lowering the feature maps' spatial dimensions to create an all-encompassing using equation (16).

$$Q_{g,v} = \sum_{m=1}^{G} \sum_{n=1}^{V} \hat{v}_{m,n}^{Q_{g,v}} W_{m,n} \tag{16}$$

Where, $\hat{v}_{m,n}^{Q_{g,v}}$ indicates the average pooling function in fabric images and $W_{m,n}$ represents the fully connected layer. After going through fully connected layers for processing, where more high-level patterns are learned, this vector is then passed via the layer for detection. The output layer provides probability or defect ratings using softmax activation function, depending on the task using equation (17).

$$v_{m,n}^{Q_{g,v}} = \left(\frac{1}{\sqrt{N_o}}\right)\left(O_{g,v}.L_{m,n} + O_{g,v}.R_{m-g}^{G} + O_{g,v}.R_{n-v}^{V}\right) \tag{17}$$

Where, R_{m-g}^{G} indicates the fabric function in hidden layer and R_{n-v}^{V} represents the probability of each image pixel. The dense activation layer of FA-ResNet is included, and it is used in place of the convolutional layer of AARCNN. To use FA-ResNetZhan et al. (2023)Dalirinia et al. (2024), the AARCNN convolutional layer with 1000 neurons replaced with two neurons. The new fully connected layer with 256 neurons is followed by dense layer. The addition of a dropout layer using 0.4 values prevented over fitting. The 1000-neuron convolutional layer of the AARCNN was replaced to create FA-ResNet. To assign expected labels to inputs, the top layer employed the Softmax layer. Graph convolution is combined with FA-ResNet techniques to capture complex linkages along longer range interactions among the fabric, as shown in eqn (18).

$$\sigma = \frac{1}{2d \times L \times N} \sum_{i=1}^{N} \sum_{j=1}^{L} \left(\left\{\hat{E}_{ij}\right\} - v_i\right)^2 \tag{18}$$

Where, σ indicates the spatial size of images;dindicates the 2d dimensional pixel in each layers;Lindicates the length of each pixel compares to the neighbor pixel;Nimplies total count of detected pixel layers;$\hat{E}_{ij}$ indicates the long rage interaction between each images and $v_i$is the variance sample of fabric images.

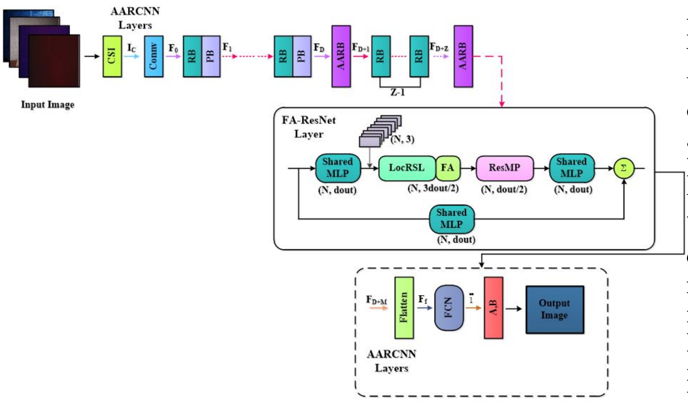

Fig. 2. Display the architecture of AARCNN-FA-ResNet

The feature maps' spatial dimensions are reduced at this point. The feature vector is sent into fully connected layers after AARCNN for advanced pattern recognition and classification. An output layer that employs a softmax or sigmoid activation function to categorize the fabric as captured, hole, horizontal and vertical fabrics to pinpoint particular defect kinds comes after these dense layers using equation (19).

$$\{h_{ij}\} = \eta \otimes \frac{\left\{\hat{E}_{ij}\right\} - v_i}{\sigma + \varepsilon} + \mu \tag{19}$$

Where, h_{ij} represents the dense layer;η represents the exact pixel location of detected fabric images;$\otimes$ denotes the activation function of AARCNN-FA-ResNet and εandμindicates the learnable parameter. Finally, AARCNN-FA-ResNet detected fabric defect like Captured, Hole, Horizontal and Vertical. The AARCNN-FA-ResNet classifier combines an AI-based optimization technique because of its expediency and relevance. Here, LEOA is utilized to optimize the AARCNN-FA-ResNet. LEOA is used for tuning the weight and bias parameter of the AARCNN-FA-ResNet.

Specifically, the close-loop represents the integration of both local (LocRSL) and global (ReMP) feature refinement stages. The loop enables residual connections that propagate enhanced features through shared MLPs and attention-based refinement blocks, ensuring richer contextual learning and preserving spatial details. The mechanism improves convergence and robustness in detecting complex fabric defects

3.5 Optimization using Lotus Effect Optimization Algorithm

The LEOA is used to improve weights parameters $\hat{E}_{ij}$ and $\hat{v}_{m,n}^{Q_{g,v}}$ of proposed AARCNN-FA-ResNet. The parameter $\hat{E}_{ij}$is implemented for enhancing the accuracy and $\hat{v}_{m,n}^{Q_{g,v}}$reducing the computation time.Water collects on plant's leaves and slides down them without being absorbed due to a characteristic of the leaves. The way the drips are moving on the leaves in this instance might be regarded as a local search. The proposed LEOA is inspired by the qualities of the Lotus flower and it pollinates. The effectiveness of the dragonfly algorithm which is made possible through pollination and local search involving water movement on lotus flower leaves is the foundation for the development of this method. The effectiveness of the dragonfly algorithm, which is made possible by pollination and a local search by water movement on lotus flower leaves, served as the foundation for the development of this technique. The two primary stages of pollination like non-biological process of self-pollination and biological process of cross-pollination are represented by the LEOA exploration and extraction phases. Here, a detailed procedure for acquiring suitable AARCNN-FA-ResNet values using LEOA is provided. Moreover, it generates a populace that is uniformly dispersed to optimize the optimal AARCNN-FA-ResNet parameters. The step method in its entirety is then displayed below;

Step 1: Initialization Randomness initially creates the LEOA's initial populace. The initialization is obtained in equation (20).

$$B(a) = \begin{bmatrix} B_1 = B(a_1) \\ B_i = B(a_i) \\ B_n = B(a_n) \end{bmatrix}_{n\times 1} \quad (20)$$

Where, $B(a)$denotes the current individual at the evolution iterationTwith index alsonsignifies the population density of neighborhoods.

Step 2: Random generation Input weight parameter $\hat{E}_{ij}$and$\hat{v}_{m,n}^{Q_{g,v}}$created at random using LEOA method.

Step 3: Fitness function Fitness function generates a random solution from initialization. It is assessed through parameter optimization using equation (21).

$$FitnessFunction = optimizing[\hat{E}_{ij} and \hat{v}_{m,n}^{Q_{g,v}}] \quad (21)$$

Where, $\hat{E}_{ij}$ is employed for increasing the accuracy,$\hat{v}_{m,n}^{Q_{g,v}}$ used for lessening computation time.

Step 4: Exploration Phase $[\hat{E}_{ij}]$ The extraction phase in the suggested method is local pollination, also known as self-fertilization. By this, the size of each bloom's growth zone around flower that has the highest chance of pollination determined by a coefficient called LEOA. The initial solution is optimal LEOA solution and all subsequent solutions converge on it. The movement algorithm's steps are performed slowly in the beginning and more rapidly in the end. Then the location of pollen is given by equation (22).

$$B_i^{T+1} = B_i^T + q\left(B_i^T - p^*\right).\hat{E}_{ij} \quad (22)$$

Where, B_i^{T+1} denotes the location in$(T+1)^{th}$ iteration;B_i^T signifies alignment of i^{th}separate in the lotus; $p^*$is the most successful pollen discovery across all known evolutionary iterations;$\hat{E}_{ij}$indicates the long rage interaction between each image andqimplies growth area, which gets smaller as the process iterates. The proposed LEOA is inspired by the qualities of the Lotus flower and its ability to pollinate. This algorithm's foundation is the effectiveness of the dragonfly algorithm, which is made possible by pollination also a localized search on the leaves of lotus flowers that involves water movement. The two primary methods of pollination are the non-biological method and the biological method, which represent the LEOA's exploring phases using equation (23).

$$q = 2f^{-\left(\frac{\Delta T}{K}\right)^C} \quad (23)$$

Where, Δ Tis the algorithm's current iteration under development; Kis denoted as the ultimate iteration number;fis changed based on the algorithm's repetition count and manages the transition between large and tiny movement steps and cindicates the each moments of LEOA.

The Lotus Effect Optimization Algorithm (LEOA) employed in this study is inspired by the natural lotus leaf surface's self-cleaning and water-repellent properties, which facilitate the algorithm's ability to escape local optima and converge toward a global solution. In the context of our model, LEOA is utilized to optimize the weight parameters of the AARCNN-FA-ResNet classifier to maximize classification accuracy.

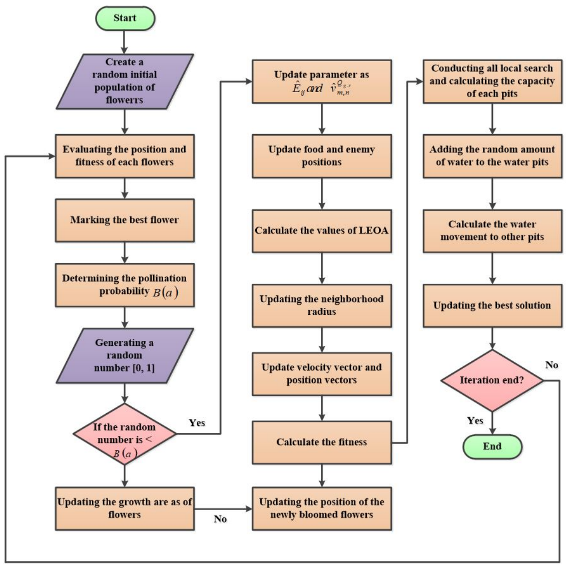

Fig. 3. Flow Chart of LEOA for Optimizing AARCNN-FA-ResNet

Step 5: Exploitation phase of optimizing$[\hat{v}_{m,n}^{Q_{g,v}}]$**:** To replicate the water flowing over the lotus leaves using water droplets, a local search is given consideration. The water will overrun a leaves if the water droplets are directed towards the initial pits above the leaf. In LEOA, the deepest pit is regarded as valuable pit. In the local search modeling, the starting value of the velocity vector for each drop is length of primitive step derived using the input. The velocity vector in the LEOA is increased by the motion vector, which is increased by the velocity vector of eqn (24).

$$y_i^T = \frac{\left(\left|e_i^T - e_{Max}\right|\right) \times Const}{\left(\left|e_{Min} - e_{Max}\right|.\hat{v}_{m,n}^{Q_{g,v}}\right)} \quad (24)$$

Where,y_i^T is the pit's capacity i^{th} in the current evolutionary iteration(T); e_i^Tsignifies the size of the evolution iteration's i^{th} pit; $e_{Max}$is the largest fitness among the pits in terms of size and is the dimensions of the pit with the smallest fitness;$\hat{v}_{m,n}^{Q_{g,v}}$indicates the average pooling function in fabric images and Constis a constant value denoting the largest objective function that a pit can accommodate. Here, the deepest pit that is, the pit with the highest fitness is considered to be the most valuable. Given the input, the distance of the primitive step establishes the starting value of the velocity vector assigned to each drop in the local search modeling; the movement vector uses equation (25) to increase its velocity vector.

$$Select_i^T = \frac{y_i^T}{\sum_{i=0}^{l} y_j^T} \quad (25)$$

Where,$Select_i^T$is the likelihood that a pit will be chosen during evolution iteration$(T)$and lis the quantity of pits

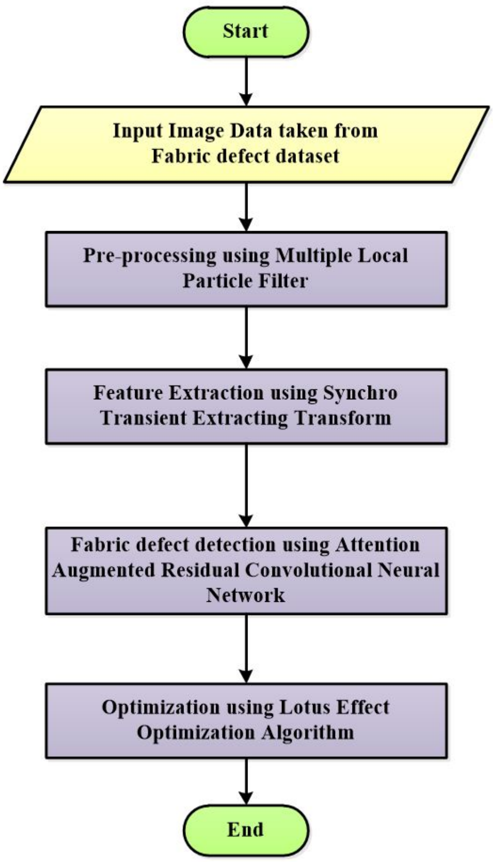

Fig. 4. Display the work flow chart for of AARCNN-FA-ResNet-FDD proposed method.

whose capacity is greater than the capacity of the spilled pit.

Step 6: Termination The weight parameter value of generator $\hat{E}_{ij}$and$\hat{v}_{m,n}^{Q_{g,v}}$ from AARCNN-FA-ResNet is enhanced through LEOA; and will repeat step 3 until satisfy the halting criteria$B\,(a) = B\,(a) + 1$.Then AARCNN-FA-ResNet-FDD imperfectly assesses the quality of analyzing the detection of fabric defect by greater accuracy, lessening computation time with error. Figure 4 indicates the Display the work flow chart for of AARCNN-FA-ResNet-FDD proposed method.

4. RESULT AND DISCUSSION

The simulation outcomes of the proposed AARCNN-FA-ResNet-FDD are discussed. The proposed method is simulated using Python on PC using an Intel Core i5 2.50GHz CPU, 8GB RAM, Windows 7, with Fabric Defect Dataset. The AARCNN-FA-ResNet-FDD model is examined under several performance metrics. Attained result of AARCNN-FA-ResNet-FDD method is analyzed with existing techniques like FSDC-CSO-DRN [21], LSTM-TC-FDD [22] and FN-Net: LCNN-FDD-TCD [23] systems .Fig 4 displays the output of the AARCNN-FA-ResNet-FDD. The AARCNN-FA-ResNet-FDD model was trained and evaluated using Tensor Flow framework with GPU acceleration disabled. The input images used for training and evaluation were obtained from the publicly available Fabric Defect Dataset, which is provided by the Intelligence Lab, Department of Computer Science and Engineering, University of Moratuwa, and hosted on Kaggle. The performance of the proposed model was evaluated using widely accepted standard metrics in the field of image classification and machine learning — including accuracy, precision, sensitivity (recall), specificity, F1-score, ROC, and computation time — as defined in international literature [e.g., IEEE, Springer, Elsevier studies]. These metrics allow consistent comparison with existing methods in related works."

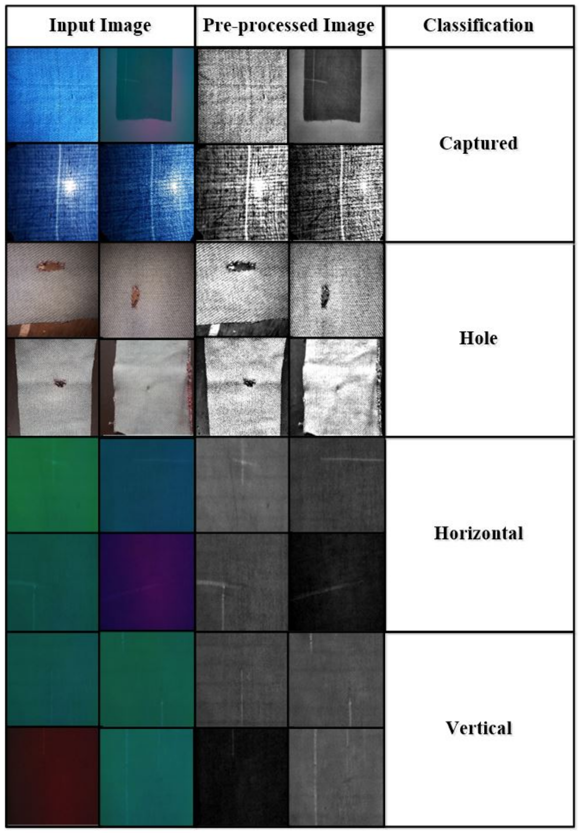

Fig. 5. Display the output of AARCNN-FA-ResNet-FDD method

4.1 Performance measures

The performance of the proposed method is examined using the metrics such as accuracy, precision, sensitivity time, ROC.

Accuracy Accuracy describes detection rate that are accurately analyze fabric defect detection on fabric surface. It is calculated using equation (26).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{26}$$

Where, TP represents True Positive; TN represents True Negative; FP represents False Positive, FN represents False Negative.

Precision It measures the positive result count though analyzes of the fabric defect detection on fabric surface using equation (27).

$$Precision = \frac{TP}{(TP + FP)} \tag{27}$$

F1-score The F1-score is analyzed and measured using equation (28).

$$F1 - Score = 2 \times \frac{recall \times precision}{recall + precision} \tag{28}$$

Sensitivity Sensitivity is commonly used in binary classification tasks based on fabric surface to correctly identify the fabric defect detection and computed using equation(29).

$$Sensitivity = \frac{TN}{(FP + TN)} \tag{29}$$

Specificity It approximates the proportion of negative cases as expressed in equation (30).

$$Specificity = \frac{TP}{FN + TP} \tag{30}$$

ROC It is the ratio of FN to TP areas is formulated using equation (31).

$$ROC = 0.5 \times \left(\frac{TP}{TP + FN} + \frac{TN}{TN + TP}\right) \tag{31}$$

Computation Time The term "computation time" describes how long a computer or algorithm needs to run a certain operation, computation, or activity. Depending on the size and complexity of the task, it is a crucial indicator of performance and efficiency in computing systems and is frequently stated in seconds, milliseconds, microseconds, or nanoseconds.

4.2 Performance metrics

Figure 5-11 shows simulation result of AARCNN-FA-ResNet-FDD method. The performance metrics are analyzed with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods

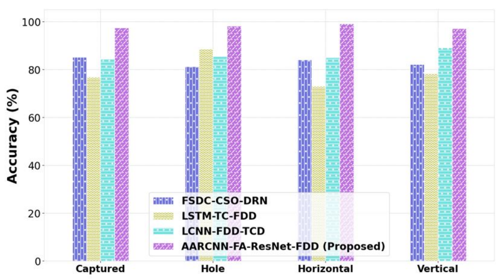

Fig. 6. Accuracy analysis

Figure 6 indicates accuracy analysis. The graph provides classification accuracy of various models for the Captured, Hole, Horizontal and Vertical attitudes associated with fabric surface influence on education is compared. Below is an analysis of the graph: Axis Vertical Y-axis displays accuracy as a percentage (%) between 0 and 100. The

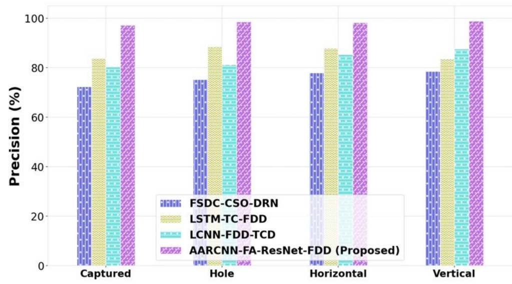

Fig. 7. Precision analysis

accuracy % of each model in identifying fabric defects is displayed on the graph. This demonstrates that the proposed AARCNN-FA-ResNet-FDD performs better in every fabric defect detection as, 16.43%, 24.29% and 30.82% higher accuracy for Captured; 18.21%, 22.96% and 30.77% higher accuracy for Hole; 15.44%, 24.90% and 31.30% higher accuracy for Horizontal and 18.64%, 26.12% and 34.49% higher accuracy for Vertical compared to existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods.

Figure 7 indicates precision analysis. The graph displays the precision (%) for Captured, Hole, Horizontal and Vertical for each model. Across all fabric defect detection categorized and the proposed model AARCNN-FA-ResNet-FDD achieves the greatest precision rates. Nearing precision, LSTM-TC-FDD demonstrates strong performance, particularly in fabric defects. Though it still does rather well in fabric surface FSDC-CSO-DRN typically has the lowest precision of all the models. This demonstrates that the proposed AARCNN-FA-ResNet-FDD performs better in every fabric defect detection as 18.64%, 24.88% and 31.95%higher precision for Captured; 16.43%, 24.29% and 30.82% higher precision for Hole; 17.05%, 24.43% and 35.89% higher precision for Horizontal and 15.09%, 23.41% and 31.86% higher precision for Vertical which is analyzed with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods.

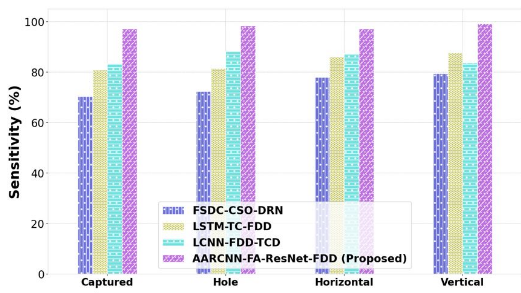

Fig. 8. Sensitivity analysis

Figure 8 indicates the sensitivity. The sensitivity percentage gauges the techniques accuracy in locating pertinent examples of each fabric defect type represented by the vertical Y-axis. The range of the scale is 0% to 100%. Three categories as Captured, Hole, Horizontal and Vertical represent the various fabric surface kinds that are represented by the horizontal x-axis. All fabric types consistently provides the highest sensitivity percentages, with the AARCNN-FA-ResNet-FDD Proposed model hitting 99.7 % in every instance. In all sentiment

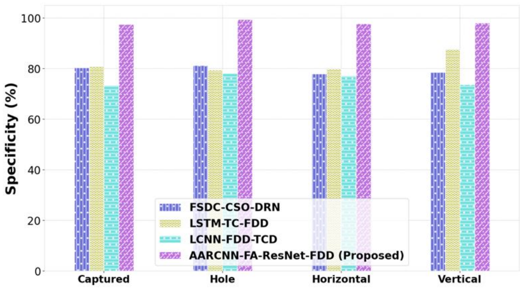

Fig. 9. Specificity analysis

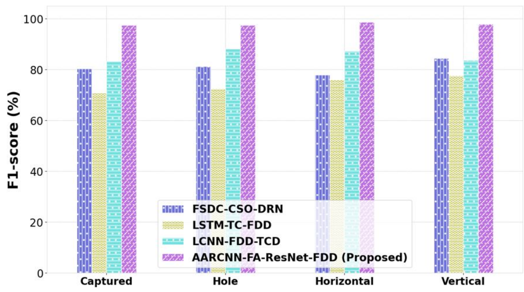

Fig. 10. F1-Score analysis

kinds, FSDC-CSO-DRN outperforms LSTM-TC-FDD and LCNN-FDD-TCD, but it falls short of the proposed model. It also demonstrates the other models outperform them despite their moderate performance. This demonstrates that the proposed AARCNN-FA-ResNet-FDD performs better in every fabric defect detection as, 19.84%, 24.93% and 31.62% higher sensitivity for Captured; 16.60%, 20.06% and 25.35% higher sensitivity for Hole; 19.45%, 24.86% and 30.97% higher sensitivity for Horizontal and 32.62%, 30.60% and 17.85% higher sensitivity for Vertical which is analyzed with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods.

Figure 9 represents specificity analysis. The given graph contrasts the specificity of numerous methods for categorizing sentiments into Captured, Hole, Horizontal and Vertical categories. The sorts of fabric defects are represented on the x-axis. Specificity is represented as percentage on y-axis with values ranging from 0% to 100%. Specificity is a measure of how well a method detects true negatives, or occurrences of non-target classes, and is given as a percentage. The method's ability to prevent false positives improves with increasing specificity. This approach is the most dependable since it shows the highest efficacy in correctly identifying various fabric defect detection categories with the fewest false positives. This demonstrates that the proposed AARCNN-FA-ResNet-FDD performs better in every fabric defect detection as, 17.89%, 23.71% and 32.82%higher specificity for Captured; 35.60%, 30.41% and 18.75% higher specificity for Hole; 26.89%, 29.49% and 18.64% higher specificity for Horizontal and 26.80%, 14.41% and 26.60% higher specificity for Vertical which is analyzed with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods.

Figure 10 indicates f1-score analysis. F1-score as the assessment metric, the graph compares the performance of fabric defect detection models for each fabric categories as Captured, Hole, Horizontal and Vertical. In every fabric categories the proposed AARCNN-FA-ResNet-FDD model performs better than the other three models FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD. Compared to the other models, the AARCNN-FA-ResNet-FDD model exhibits much higher F1-scores in every category, indicating its superior efficacy in fabric surface. This demonstrates that the proposed AARCNN-FA-ResNet-FDD performs better in every fabric defect detection as, 15.60%, 32.42% and 26.21% higher F1-Score for Captured; 30.20%, 16.51% and 30.90% higher F1-Score for Hole; 14.92%, 25.75% ,32.18% greater F1-Score for Horizontal and 22.60%, 20.96% and 15.39% greater F1-Score for Vertical compared to existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods.

Figure 11 indicates the computation time. The graph provides the contrast calculation times of several techniques employed in a fabric defect detection study on fabric surface. The AARCNN-FA-ResNet-FDD approach is the most efficient of the methods compared since it has the lowest computing time. The longer computation periods for LSTM-TC-FDD and LCNN-FDD-TCD indicate that they are less effective. This graph demonstrates the proposed AARCNN-FA-ResNet-FDD method's computational efficiency, which makes it a superior option for applications in fabric settings where speedy processing is essential. This demonstrates that the proposed AARCNN-FA-ResNet-FDD performs better in every fabric defect detection as, 32.22%, 32.60% and 27.05% lower computation time for Captured, Hole, Horizontal and Vertical which is analyzed with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCDmethods.

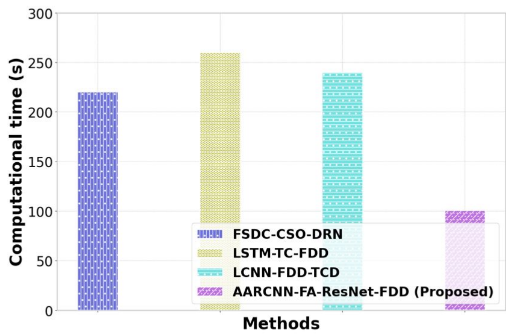

Fig. 11. Computational Time analysis

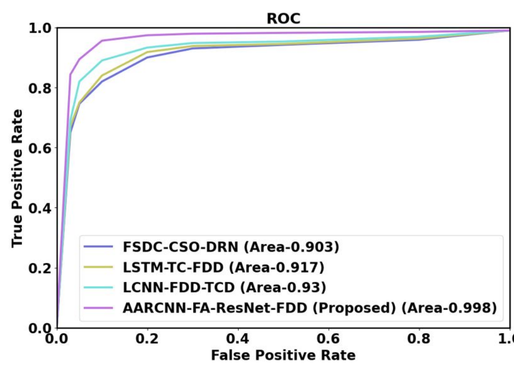

Fig. 12. ROC analysis

Fig 12 indicates the ROC. This model's capacity to discriminate among the classes is indicated by the ROC value.

The model performs better the closer its ROC value is near 1. With the greatest ROC in this graph, AARCNN-FA-ResNet-FDD performs the best out of all the models that were compared. The y-axis is the percentage of TPs that the technique accurately identified. X-axis is the percentage of actual negatives that are misinterpreted as positives. It indicates that AARCNN-FA-ResNet-FDD model performs better with every fabric defect detection as 0.90%, 0.91% and 0.93% higher ROC for Captured, Hole, Horizontal and Vertical analyzed with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCDmethods.

4.3 Discussion

A novel AARCNN-FA-ResNet-FDD technique to fabric defect detection on fabric surface using Fabric Defect Dataset is developed in this paper. The AARCNN-FA-ResNet-FDD model encompasses MLPF based preprocessing; STET based feature extraction and AARCNN-FA-ResNet based fabric defect detection on fabric surface classification Captured, Hole, Horizontal and Vertical. Finally, AARCNN-FA-ResNettechnique is used for performing classification that detects the fabric defect detection in fabric surface. In the Fabric Defect Dataset, the approach's highest average results are compared with the average results of existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD techniques. This is less expensive than the proposed method. The proposed method for premature data enhancement undergoes classification of data classification for fabric defect detection on fabric surface from Fabric Defect Dataset. The proposed approach combines the LEOA algorithm with a quicker AARCNN-FA-ResNet, which improves data collecting efficiency and more successfully tackles the problem of model over fitting. The proposed AARCNN-FA-ResNet-FDD attains accuracy of 18.21%, 22.96%, 30.77% higher than existing like FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods. In contrast, the accuracy of the proposed AARCNN-FA-ResNet-FDD is 97.57% when compared to 88.75% accuracy and precision of comparison methodologies. The proposed AARCNN-FA-ResNet-FDD strategy has better evaluation criteria for accuracy and precision when compared to current techniques. Thus the proposed method is less expensive than the comparative methods. Consequently, the proposed method more effectively and efficiently detects early fabric defect detection on fabric surface.

5. CONCLUSION

The AARCNN-FA-ResNet-FDD is successfully executed in this research. The proposed AARCNN-FA-ResNet-FDD method is simulated in Python utilizing Fabric Defect Dataset. Several experiments are performed to posts and comment the fabric defect detection on fabric surface. Then the image is extracted from Fabric Defect Dataset, for classifying the Fabric Defect on fabric surface as Captured, Hole, Horizontal and Vertical. The AARCNN-FA-ResNet-FDD performed better with the Co-training technique than used separately with Recall, Specificity, F1-score, ROC. The performance of Captured, Hole, Horizontal and Vertical approach attains 19.84%, 24.93% and 31.62% higher sensitivity; 17.89%, 23.71% and 32.82% higher specificity;15.60%, 32.42% and 26.21%higher F1-Score and 0.90%, 0.91% , 0.93% greater ROC compared with existing FSDC-CSO-DRN, LSTM-TC-FDD and LCNN-FDD-TCD methods. It is planned to use a CNN in a combination of methods in the future to validate the model with multiple datasets and increase accuracy. The dataset can be altered by creating artificial images of different flaws with backgrounds that have similar textures, and more network testing is anticipated. To satisfy the demands of the fabric industry, the model also needs more hybrid optimization.

REFERENCES

Althubiti, S.A., Alenezi, F., Shitharth, S., K, S., and Reddy, C.V.S. (2022). Circuit manufacturing defect detection using vgg16 convolutional neural networks. *Wireless Communications and Mobile Computing*, 2022(1), 1070405.

¸C. Seçkin, A. and Seckin, M. (2022). Detection of fabric defects with intertwined frame vector feature extraction. *Alexandria Engineering Journal*, 61(4), 2887–2898.

Chen, M., Yu, L., Zhi, C., Sun, R., Zhu, S., Gao, Z., Ke, Z., Zhu, M., and Zhang, Y. (2022). Improved faster r-cnn for fabric defect detection based on gabor filter with genetic algorithm optimization. *Computers in Industry*, 134, 103551.

Cheng, L., Yi, J., Chen, A., and Zhang, Y. (2023). Fabric defect detection based on separate convolutional unet. *Multimedia Tools and Applications*, 82(2), 3101–3122.

Dalirinia, E., Jalali, M., Yaghoobi, M., and Tabatabaee, H. (2024). Lotus effect optimization algorithm (lea): a lotus nature-inspired algorithm for engineering design optimization. *The Journal of Supercomputing*, 80(1), 761–799.

Fang, B., Long, X., Sun, F., Liu, H., Zhang, S., and Fang, C. (2022). Tactile-based fabric defect detection using convolutional neural network with attention mechanism. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–9.

Fouda, Y. (2022). Integral images-based approach for fabric defect detection. *Optics & Laser Technology*, 147, 107608.

Gao, Y., Gao, L., and Li, X. (2023). A hierarchical training-convolutional neural network with feature alignment for steel surface defect recognition. *Robotics and Computer-Integrated Manufacturing*, 81, 102507.

Jain, S., Seth, G., Paruthi, A., Soni, U., and Kumar, G. (2022). Synthetic data augmentation for surface defect detection and classification using deep learning. *Journal of Intelligent Manufacturing*, 1–14.

Jha, S.B. and Babiceanu, R.F. (2023). Deep cnn-based visual defect detection: Survey of current literature. *Computers in Industry*, 148, 103911.

Kaggle User: rmshashi (2022). Fabric defect dataset. `https://www.kaggle.com/datasets/rmshashi/fabric-defect-dataset`. Accessed: 2025-04-29.

Kumar, K.S. and Bai, M.R. (2023). Lstm based texture classification and defect detection in a fabric. *Measurement: Sensors*, 26, 100603.

Li, T., Sbarufatti, C., and Cadini, F. (2024). Multiple local particle filter for high-dimensional system identification. *Mechanical Systems and Signal Processing*, 209, 111060.

Li, W., Zhang, H., Wang, G., Xiong, G., Zhao, M., Li, G., and Li, R. (2023). Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robotics and Computer-Integrated Manufacturing*, 80, 102470.

Liu, G. and Li, F. (2022). Fabric defect detection based on low-rank decomposition with structural constraints. *The Visual Computer*, 38(2), 639–653.

Liu, Q., Wang, C., Li, Y., Gao, M., and Li, J. (2022). A fabric defect detection method based on deep learning. *IEEE Access*, 10, 4284–4296.

Lu, L., Hou, J., Yuan, S., Yao, X., Li, Y., and Zhu, J. (2023). Deep learning-assisted real-time defect detection and closed-loop adjustment for additive manufacturing of continuous fiber-reinforced polymer composites. *Robotics and Computer-Integrated Manufacturing*, 79, 102431.

Ma, Y., Yu, G., Lin, T., and Jiang, Q. (2024). Synchro-transient-extracting transform for the analysis of signals with both harmonic and impulsive components. *IEEE Transactions on Industrial Electronics*.

Mahmood, T., Ashraf, R., and Faisal, C.N. (2023). An efficient scheme for the detection of defective parts in fabric images using image processing. *The Journal of The Textile Institute*, 114(7), 1041–1049.

Meeradevi, T. and Sasikala, S. (2024). Automatic fabric defect detection in textile images using a labview based multiclass classification approach. *Multimedia Tools and Applications*, 1–20.

Mewada, H., Pires, I.M., Engineer, P., and Patel, A.V. (2024). Fabric surface defect classification and systematic analysis using a cuckoo search optimized deep residual network. *Engineering Science and Technology, an International Journal*, 53, 101681.

Pourkaramdel, Z., Fekri-Ershad, S., and Nanni, L. (2022). Fabric defect detection based on completed local quartet patterns and majority decision algorithm. *Expert Systems with Applications*, 198, 116827.

Shao, L., Zhang, E., Ma, Q., and Li, M. (2022). Pixel-wise semisupervised fabric defect detection method combined with multitask mean teacher. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–11.

Singh, S.A. and Desai, K.A. (2023). Automated surface defect detection framework using machine vision and convolutional neural networks. *Journal of Intelligent Manufacturing*, 34(4), 1995–2011.

Suryarasmi, A., Chang, C.C., Akhmalia, R., Marshallia, M., Wang, W.J., and Liang, D. (2022). Fn-net: A lightweight cnn-based architecture for fabric defect detection with adaptive threshold-based class determination. *Displays*, 73, 102241.

Wang, P., Yang, Y., and Moghaddam, N.S. (2022). Process modeling in laser powder bed fusion towards defect detection and quality control via machine learning: The state-of-the-art and research challenges. *Journal of Manufacturing Processes*, 73, 961–984.

YaşarÇıklaçandır, F.G., Utku, S., and Özdemir, H. (2024). Determination of various fabric defects using different machine learning techniques. *The Journal of The Textile Institute*, 115(5), 733–743.

Yu, X., Lyu, W., Zhou, D., Wang, C., and Xu, W. (2022). Es-net: Efficient scale-aware network for tiny defect detection. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–14.

Zhan, L., Li, W., and Min, W. (2023). Fa-resnet: Feature affine residual network for large-scale point cloud segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 118, 103259.

Zhang, B., Sifaou, H., and Li, G.Y. (2023). Csi-fingerprinting indoor localization via attention-augmented residual convolutional neural network. *IEEE Transactions on Wireless Communications*.

Zhao, H., Wang, J., Li, C., Liu, P., and Yang, R. (2024). Fabric defect detection via feature fusion and total variation regularized low-rank decomposition. *Multimedia Tools and Applications*, 83(1), 609–633.

Zhao, S., Zhong, R., Wang, J., Xu, C., and Zhang, J. (2023). Unsupervised fabric defects detection based on spatial domain saliency and features clustering. *Computers & Industrial Engineering*, 185, 109681.

Üzen, H., Turkoglu, M., Aslan, M., and Hanbay, D. (2023). Depth-wise squeeze and excitation block-based efficient-unet model for surface defect detection. *The Visual Computer*, 39(5), 1745–1764.