Heliyon 10 (2024) e26466

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Image synthesis of apparel stitching defects using deep convolutional generative adversarial networks

Noor ul-Huda [a], Haseeb Ahmad [a,*], Ameen Banjar [b], Ahmed Omar Alzahrani [b], Ibrar Ahmad [c], M. Salman Naeem [d]

[a] *Department of Computer Science, National Textile University, Faisalabad, Pakistan*
[b] *College of Computer Science and Engineering, University of Jeddah, 21959, Jeddah, Saudi Arabia*
[c] *Department of Computer Science, University of Peshawar, Peshawar, Pakistan*
[d] *Department of Textile Engineering, National Textile University, Faisalabad, Pakistan*

ARTICLE INFO

Index terms:
Deep learning
Deep convolutional generative adversarial network
Defect generation
Image synthesis

ABSTRACT

In industrial manufacturing, the detection of stitching defects in fabric has become a pivotal stage in ensuring product quality. Deep learning-based fabric defect detection models have demonstrated remarkable accuracy, but they often require a vast amount of training data. Unfortunately, practical production lines typically lack a sufficient quantity of apparel stitching defect images due to limited research-industry collaboration and privacy concerns. To address this challenge, this study introduces an innovative approach based on DCGAN (Deep Convolutional Generative Adversarial Network), enabling the automatic generation of stitching defects in fabric. The evaluation encompasses both quantitative and qualitative assessments, supported by extensive comparative experiments. For validation of results, ten industrial experts marked 80% accuracy of the generated images. Moreover, Fréchet Inception Distance also inferred promising results. The outcomes, marked by high accuracy rate, underscore the effectiveness of proposed defect generation model. It demonstrates the ability to produce realistic stitching defective data, bridging the gap caused by data scarcity in practical industrial settings.

1. Introduction

The textile industry is indeed one of the largest industries in the world, producing millions of tons of fabric every year, plays a significant role in the global economy [1]. The production of fabrics is now a highly optimized process due to advancement of technology but it may contain some defects. Fabric defects are a significant issue that can affect product quality, customer satisfaction, and overall profitability [2].

Fabric defect detection is a crucial step in industrial manufacturing to ensure product quality and avoid negative consequences. If defects go unnoticed, it can create biased results in the quality assessment. This can lead to defective products reaching the market, harms both the company's reputation and the economic gains. Thus, detecting defects is immensely important for maintaining product quality, enhancing production efficiency [3]. In the textile industry, identifying faulty sections within fabric traditionally involves a human inspection process that is both time-consuming and expensive. For automated identification and detection of these defects we need deep learning models.

* Corresponding author
E-mail address: haseeb_ad@hotmail.com (H. Ahmad).

https://doi.org/10.1016/j.heliyon.2024.e26466
Received 3 October 2023; Received in revised form 12 February 2024; Accepted 14 February 2024
Available online 15 February 2024

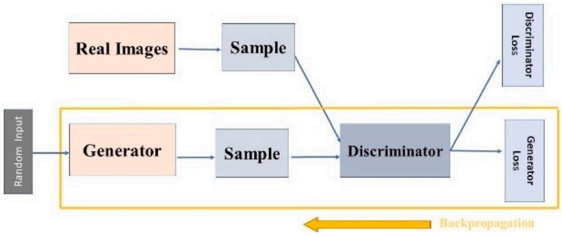

Fig. 1. Training of GAN generator [15].

For the purpose of detecting fabric defects, numerous image processing models have been used. An improved technique utilizes membership degrees of fabric regions and applies an iterative threshold approach to ensure precise and accurate detection. Genetic algorithms are employed to optimize Gabor filters, achieving accurate defect detection in patterned fabrics [4]. a novel sequential defect detection method designed for patterned fabrics [5]. Traditional methods for feature extraction often involve manually designed extractors, required expertise and complex adjustments. These methods were customized to specific applications, resulting in inadequate generalization and robustness [6].

To address the challenges in fabric defect detection, many researchers have turned to neural networks as a promising solution. In a particular study, a compact Convolutional Neural Network (CNN) architecture designed specifically for detecting common fabric defects [7]. Their approach integrated multiple microarchitectures and a multi-layer perceptron to optimize the network, ultimately enhancing the accuracy of defect detection. Similarly, an integrated CNN model that leveraged long-short-term visual memory to extract three different types of features, enabling more effective fabric defect detection [8]. An enhanced version of Faster R–CNN (Region Convolutional Neural Network) has been introduced for the purpose of detecting defects in fabric [9]. They used the Gabor filter with genetic algorithm optimization as a basis for their approach. These neural network-based approaches offer improved potential for generalization and robustness compared to traditional methods, addressing the limitations associated with manual feature extraction.

Deep learning techniques, however, need thousands or hundreds of training data samples. The performance of deep learning neural networks in detecting fabric defects depends largely on the size and diversity of the training datasets. Generating and gathering large dataset of real-world images of appeal defects (stitching) for machine learning algorithms, can be a challenging task. Defect databases of textile images are too small and unbalanced because real world data of fabric defects are usually not captured. If the defective data is captured, it is not shared by industries because of reputation issues and privacy concerns. Huge amount of apparel defects (stitching defects) data is not available due to research-industrial linkage gap, remote industries, or due to privacy concerns of industrial stakeholders. Insufficient training data can lead to over fitting during model training, resulting in reduced detection precision. As a result, defects of stitching inspection is greatly hindered, primarily attributed due the lack of a dataset for experimental evaluation. The limited availability of apparel stitching defects data makes it difficult to train deep learning models for industrial solutions. This limitation hinders the further advancement and practical application of deep learning in fabric defect detection.

To increase training dataset, researchers have focused on data augmentation techniques. These techniques involve various transformations such as flipping, rotation, zooming, random cropping, and color adjustment, and adding noise to the training data. For example, rotating and flipping single-cell images reduced measurement time, while changing brightness and rotation improved fruit detection using Faster R–CNN [10]. Methods like flipping, Gaussian blur, and brightness changes have enhanced the performance of deep convolutional neural networks (DCNN) in radiation and image classification tasks [11]. Tailoring and filtering techniques have also been employed to enhance data for accurate lung cancer testing [12]. Generating data using augmentation techniques leads to over fitting because it changes the orientation of same object. These techniques can provide some variation in the data, they only produce a limited amount of alternative data. However it's crucial to acknowledge that these techniques mainly focus on altering the existing data rather than enhancing the image information, and their generalization effect may be limited due to certain constraints [6].

An approach used to generate data of (stitching) defects of appeal is synthetic data generation [13]. In recent years, synthetic data production has become more significant in a variety of applications. In this research, we intend to generate synthetic data of defects (stitching) of apparel using GAN [14]. By generating synthetic data that represents a variety of defects, researchers can train and test machine learning models to accurately identify and classify defects in real-world fabric images. Artificially created synthetic data using GAN provide the chance to enhance the current situation by producing defect-related data with properties similar to those of the real-world data.

Image synthesis refers to the process of generating artificial images using algorithms, mathematical models, or machine learning techniques often driven by artificial intelligence (AI) and machine learning (ML). Defect generation is an application of image synthesis. It refers to the process of artificially creating or simulating defects in products, or images for various purposes, such as training and testing in quality control, machine learning, or computer vision applications.

Generative Adversarial Network (GAN) typically consists of two neural networks: a generator and a discriminator. The generator network takes as input a random noise vector and produces a new sample of data that is similar to the training data. The discriminator receives as input both the image produced by the generator and a sample of actual images and try to differentiate between generated data and real input data. It returns a probability value between 0 and 1, with a value closer to 1 denoting a greater chance that the

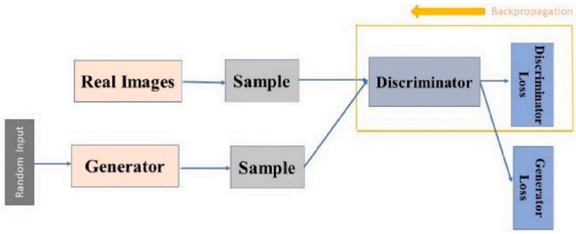

Fig. 2. Training of GAN discriminator [15].

returned image is part of the real dataset otherwise it is part of a generated sample of images [15].

Fig. 1 shows that discriminator input is directly linked to the generator output. The generator updates its weights by using the signal produced by the discriminator's classification, which is obtained through back propagation.

Fig. 2 depicts that discriminator is simply a classifier. Back propagation in discriminator training is to iteratively improve the discriminator's ability to accurately classify real and generated images.

The training of Generative Adversarial Network (GAN) has been known to be challenging and sometimes unstable. This instability can result in divergent training processes and make it difficult to achieve desired results. Training instability in GAN is due to mode collapse, where the generator fails to explore the entire data distribution and instead produces limited variations of samples. This can result in repetitive or low-diversity outputs [16].

In 2016, researchers introduced the concept of the Deep Convolutional Generative Adversarial Network (DCGAN) [17]. This approach involves utilizing Convolutional Neural Network (CNN) to effectively extract key features from datasets [18]. DCGAN aims to enhance the training process of the Generative Adversarial Network (GAN) model, making it more manageable to prevent convergence issues.

In this research, a method based on Deep Convolutional Generative Adversarial Networks (DCGAN) for image synthesis of apparel stitching defects is proposed. By creating a large dataset of synthetic images that simulate different types of stitching defects of apparel, machine learning models can be trained to accurately identify these defects. Synthetic dataset generation can be used to create a large variety of stitching defect images that can be used to test and improve defect detection algorithms. Collecting and labeling real-world images of fabric defects can be time-consuming and expensive. Synthetic dataset generation can reduce this cost through the production of a lot of images with different types and severities of stitching defects. It can help improve the accuracy of defect detection algorithm, reduce costs and support research in the textile industry.

- A preliminary dataset of stitching defects is generated for image synthesis of stitching defects.
- An effective method based on Deep Convolutional Generative Adversarial Networks (DCGAN) for image synthesis of apparel stitching defects is proposed.

The organization of the sections of this paper is shown below:

In Section 2, an in-depth review of existing literature and related work is presented. Section 3 details the methodology employed, covering the construction of the neural network model utilized in the study and outlining the data description and augmentation applied to the input data. Moving on to Section 4, the focus shifts to the experimentation phase, providing insights into the experimental setup and specific procedures followed during the experiments. Section 5 addresses the qualitative and quantitative evaluation of the generated data, introduces the criteria used for assessment and compares the results of the developed neural network with those of the pix2pix GAN neural network, employing various evaluation metrics. Finally, in Section 6, the paper concludes by summarizing key findings, showcasing results, and offering insights into the future directions and potential areas for further research in the field.

2. Related work

2.1. Defect detection

Traditional methods before deep learning: Defect recognition involves the detection and classification of surface defects through machine vision, encompassing image preprocessing, feature extraction and selection, and image recognition. Traditional fabric defect detection algorithms such as histogram analysis involve a series of steps. Initially, fabric defect images undergo preprocessing, which includes tasks like converting them to grayscale and reducing noise. Defect detection is then accomplished by identifying and categorizing these preprocessed defect images using machine learning techniques like Support Vector Machine (SVM) or Decision Trees [3]. Zhou et al. [19] utilized the projection histogram of circular regions to detect defects, employing a decision tree for defect classification. Huang et al. Tsai and Hsieh [20] applied Fourier transformations technique to detect fabric defects like scratches and oil stains.

Deep learning methods: In recent advancements in computer vision, deep learning models such as Alexnet, GoogleNet, ResNet, and DenseNet [21] have demonstrated significant progress. These technologies have proven effective in addressing challenging surface defect detection tasks. Various studies have showcased the application of deep learning techniques, such as Fisher-criterion-based models for fabric defect detection, auto-encoders for fabric surface defect identification, and deep convolutional neural networks (CNNs) for surface defects.

Despite the advantages of deep learning in fabric defect detection, a notable challenge lies in the scarcity of defect images for training in industrial production settings. Consequently, there is a recognized need to develop a method for generating synthetic fabric defects to augment the available training data for deep learning applications.

2.2. Image generation based on deep learning

Due to its excellent image processing capabilities, deep learning has becoming more prominent in image generating research. In 2016, van den Oord et al. [22] introduced Pixel RNN, a method utilizing a recurrent neural network (RNN) to predict pixel values by generating a sequence of pixels until an entire image is produced. Based on energy probability distributions, Hinton et al. and Huang et al. created images with and without labels using a restricted Boltzmann machine (RBM).

In 2013, Kingma and Welling [23]introduced the variation auto encoder (VAE), which comprises two fundamental components: an encoder and a decoder. The encoder's role is to transform real images into lower-dimensional Gaussian distributions, while the decoder performs the reverse process, translating these lower-dimensional Gaussian distributions into generated images. During the training process, the objective is to minimize the mean squared error (MSE) by comparing the generated images to real ones. Despite these approaches, they possess shortcomings like slow generation speeds and the production of blurry, low-resolution images. Consequently, these image-generation methods are ill-suited for generating industrial defect images [19].

2.3. Image based on GAN

In 2014, Goodfellow introduced the concept of the Generative Adversarial Network (GAN). GAN is a generative model that finds its theoretical basis in zero-sum game theory and is mainly utilized for the generation of images. The authors applied a min-max optimization framework to concurrently train two deep learning models: a generative model denoted as 'G' and a discriminative model referred to as 'D.' This approach is based on the concept of a zero-sum game, where the objective of one player is to maximize their gain while the other player's objective is to minimize their loss [3].

In 2018, Zhao and co-authors [24] introduced an approach by utilizing GAN networks for defect detection. Their approach includes using the GAN network to repair input defect images in order to learn defect features. By comparing defect samples with the repaired versions, they identified the defect areas within the images, achieving successful defect detection provide optimization [25–28] solution. The average detection accuracy for texture defects using this novel approach was 98.5323%, which is impressive.

A Defect Generation Network, Defect-GAN, was developed by, Zhang [29] in 2019 it can create realistic faults in image background, with unique textures and appearances. This network uses a hierarchical framework to produce defects that appear real and replicate their random variations. In the same year, Zhang et al. also put forward another method called Semi-Supervised Generative Adversarial Network (SSGAN) specifically designed for detecting defects in images. This framework included two sub-networks to enhance the overall defect detection process by increasing the precision of pixel segmentation.

Niu et al. [15]. introduced a defect image generation method that allows for control over the defect area and intensity. In this approach, the defect area is treated as an image repair task utilizing a generative adversarial network (GAN), and a defect mask is employed to regulate the defect area. By establishing a defect direction vector in the latent variable space, considering the feature continuity between defective and non-defective regions, the method achieves a one-to-many correspondence between the defect mask and the image, enabling control over defect intensity. This innovative method proves effective in enhancing both the quality and diversity of small and weak defect images, addressing the limitations associated with traditional approaches and contributing to improved defect image generation.

In recent times, Ma [30] introduced an innovative network called FusionGAN. this network is designed to efficiently merge thermal radiation information from infrared images with the textural details from visible images. Their approach successfully produces clean images without the noise issues that often arise from upscaling infrared information. Wang et al. [31] proposed an innovative technique utilizing 3D conditional GANs (3D c-GANs) to enhance low-dose positron emission tomography (PET) images to high-quality, full-dose images. Their method demonstrated superior performance compared to benchmarks and state-of-the-art techniques in both qualitative and quantitative evaluations.

A deep convolutional GAN (DCGAN) was used by Hu et al. [32] to provide an unsupervised method for detecting fabric defects. In order to generate realistic faults within defect-free samples, Liu et al. [33] used a multi-layer GAN model. Comparative tests were used to demonstrate the efficacy of their method. ManiGAN, a technique that reconstructs the original image using textual input data, was introduced by Li et al. [34].

This study presents a novel approach for addressing the shortage of industrial stitching defect images through defect image generation. Conventional defect recognition methods face two challenges: the need for distinct algorithms for different defect types and limitations in handcrafted features. Deep learning offers a solution to these issues; however, it demands a substantial volume of images, which industrial defect datasets often lack. To bridge this gap, this paper suggests a stitching defect image generation approach based on DCGAN, aimed at producing a balanced and authentic defect image dataset. In practice, the technique utilize defective training samples generated and captured by itself to generate huge number of different defect images using the DCGAN approach. This expanded dataset of stitching defects is subsequently employed to develop a deep-learning-based defect detection model.

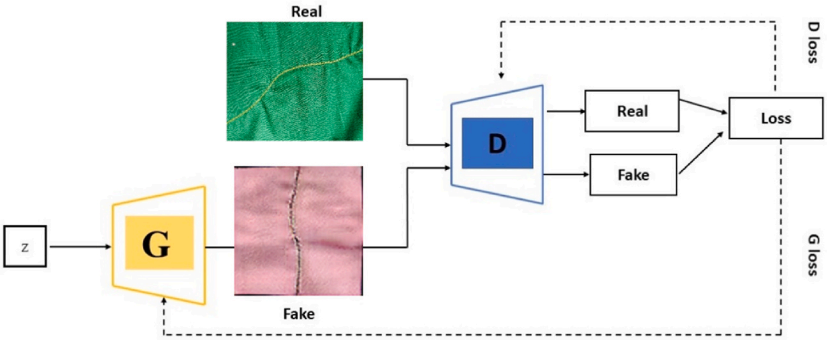

Fig. 3. GAN architecture.

3. Methodology

Stitching Defect Dataset Generation: We used an approach for stitching defect data generation called DCGAN (Deep Convolutional Generative Adversarial Network), which based on generator and discriminator model. Even with a small number of defective samples, DCGAN is capable of producing high quality and diverse stitching defect images.

3.1. Data description

The dataset comprises around 4000 images, encompassing various stitching defect scenarios in apparel. Among these, there are 1100 images illustrating unbalanced stitches, 1000 images of open seam, 1000 depicting Seam Puckering, and an additional 900 broken stitches. The dataset creation process involved the generation and capture of real-world images, simulating distinct stitching defects commonly found in garments. Initially, a diverse collection of different types of clothing was amassed. Subsequently, various colors of threads were employed during the stitching process to generate the defects. These stitching defects encompass irregularities such as unbalanced stitches, Seam Puckering, and broken stitch. The resulting images are in RGB format, each sized 256 by 256 pixels, capturing the details and characteristics of the defects. Preprocessing typically involves removing irrelevant characteristics or noise from the images. Before using the image to train our DCGAN, we also normalize the image's pixels. Some images are messy and included objects that should not be included in its respective dataset. To eliminate these, we manually removed the noisy or irrelevant images from our dataset.

3.2. Data augmentation module

When working with a limited dataset for image generation, there's a risk of overfitting during network training. Overfitting can hinder the convergence of the generated model, resulting in poor-quality images. This, in turn, impacts subsequent detection tasks. In order to solve this problem, the article suggests starting the DC-GAN network with a data augmentation module. This module aims to increase amount of original samples of defects while maintaining the integrity of the generated image distribution. By applying augmentation our original dataset is doubled. It became 6000 images approximately.

To accomplish this, three specific data augmentation techniques were employed: image scaling, image rotation, and mirroring. These techniques help diversify the dataset and reduce overfitting by introducing variations without altering the fundamental distribution of the generated images. This approach aims to improve the training process, leading to better convergence and ultimately higher-quality generated images, which in turn positively affect detection tasks.

3.3. Principle of DCGAN

In 2016, Radford et al. published a significant paper introducing Deep Convolutional Generative Adversarial Networks (DCGANs). DCGAN, an evolution of the GAN model, introduces the use of convolutional neural networks (CNNs) to extract features from source data. DCGAN enhances stability and reduces the time required for training. It incorporates Convolutional and Transposed Convolutional layers in its architecture. It builds upon the traditional GAN architecture by replacing the multi-layer perceptron (MLP) structure with a convolutional neural network (CNN) structure. In order to improve the training stability, this update involves removing the pooling layers and adding Batch Normalization between the convolutional layers and the activation functions. Fully connected hidden layers are eliminated in favor of deeper designs. The generator employs ReLU activation for internal layers and Tanh for the output, while the discriminator uses LeakyReLU throughout. These changes collectively improve training and output quality in the Generative Adversarial Network [35].

Two essential parts of DCGAN are the generative model (G) and the discriminative model (D). These two models are trained together in a simultaneous manner. The role of the discriminative model is to differentiate between genuine data and fake data samples. In contrast, the generative model is designed to learn and replicate a specific target distribution of information, effectively

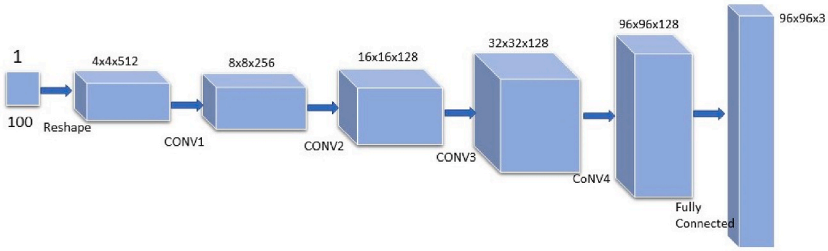

Fig. 4. Generator architecture of DCGAN model.

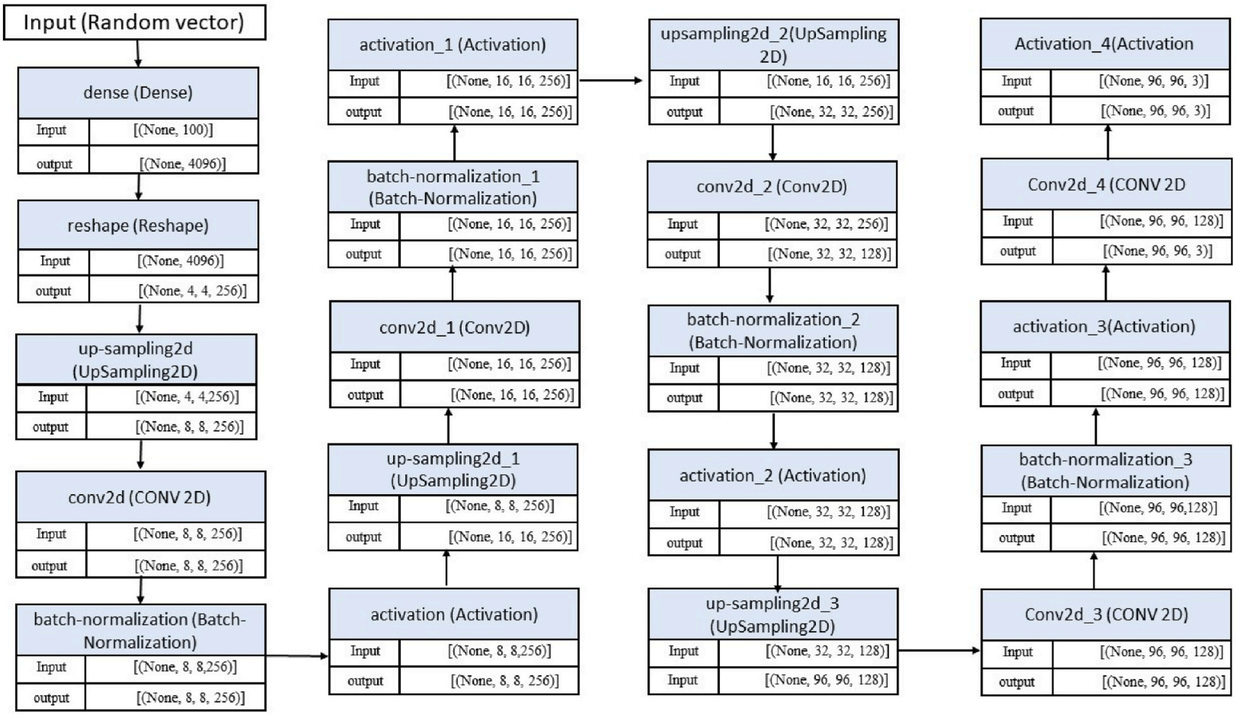

Fig. 5. Generator structure diagram.

challenging the discriminative model. The discriminative model (D) essentially acts as a binary classifier, It returns a probability value between 0 and 1, with a value closer to 1 denoting a greater chance that the returned image is part of the real dataset otherwise it is part of a generated sample of images [33].

Fig. 3 illustrates a GAN architecture where the generator uses a random noise vector to create synthetic images of unbalanced stitches. The discriminator distinguishes between real and synthetic images, minimizing the difference via a loss function. This illustrates the adversarial training dynamic between the generator and discriminator in the GAN framework.

3.4. Structure of generator network

The generator model takes 100 dimensional random noise vector that serves as the initial input for the generator and uses number of color channels in the generated images (e.g. 3 for RGB). It is defined as a Sequential model. In a Sequential model, each subsequent layer connected to the output of the previous layer. A densely connected (fully connected) layer with ReLU activation is added to the model. This layer takes the random noise vector as input and produces a high-dimensional output. The Reshape layer then reshapes this output into a $4 \times 4 \times 256$, preparing it for further processing.

The generator uses a sequence of UpSampling2D layers then Conv2D layers. UpSampling increases the spatial dimensions of the tensor, and Conv2D performs convolutional operations on it. These layers gradually transform the initial noise into a complex representation that resembles an image. To improve the stability and speed up the training process, Batch Normalization is applied to normalize the input to each unit in the network. There is a Batch Normalization layer with a momentum of 0.8. Batch normalization

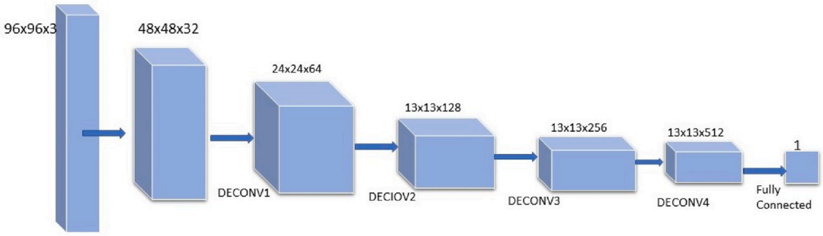

Fig. 6. Discriminator architecture of DCGAN model.

helps stabilize and accelerate training by normalizing the inputs to a layer. This normalization ensures that the input to each layer has zero mean and unit variance, which can prevent vanishing or exploding gradients during training. Fig. 4 depicts the transformation of a 100-dimensional noise vector through a series of four fractional-strided convolutional operations, resulting in a high-level representation with a spatial dimension of 96 by 96.

In the model Rectified Linear Units (ReLU) activation functions are applied to all layers except for the output layer. ReLU activation reduces negative values to zero while maintaining the integrity of positive values. This helps the network introduce non-linearity and learn complex patterns from the data.

$$\mathrm{Re}L\cup(x)=\begin{cases}x, x>0\\ 0, x\leq 0\end{cases} \tag{1}$$

Equation (1) describes the behavior of x in different intervals where it returns x for positive x and 0 for non-positive x.

There are two more rounds of upsampling and convolution operations. The purpose of these layers is to increase the spatial resolution and complexity of the tensor. The generator is intended to progressively generate higher-resolution details. The last Conv2D layer is used to produce the final generated image with kernel size of 3x3 to maintain the spatial dimensions. In the output layer, the generator utilizes the hyperbolic tangent activation function. Equation (2) represents the ratio of the difference of two exponential functions to their sum.

$$Tan(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2}$$

Tanh is a zero-mean activation function, which means its average output is centered on zero. Using Tanh in the output layer ensures that the generated data has values in the range [−1, 1], suitable for image pixel values. This normalization can enhance the training process's efficiency and stabilize the learning dynamics. Fig. 5 illustrates the structure of the generator in a visual form.

3.5. Structure of discriminator network

In the DCGAN discriminator, strided convolutions are used in place of pooling layers for the downsampling process. Strided convolutions directly reduce the spatial dimensions of the input data, which can be an effective way to downsample the data and capture important features. Similar to the generator, the model is defined as a Sequential model. This allows to stack layers sequentially.

A Conv2D layer with 32 filters (kernels) of size 3x3 serves as the foundation of the discriminator. The convolution operation is performed with a stride of 2, which means the output spatial dimensions will be reduced by half compared to the input. This layer processes the input image and extracts low-level features. Batch-normalization with a momentum of 0.80 is applied to the discriminator to improve training and reduce initialization and gradient-related problems with the model. The input to each unit in a layer is normalized by batch normalization, similar to how it is used in the generator. Batch normalization standardizes the activations of the previous layer, which contributes to a more stable training process and accelerates convergence. Fig. 6 displays the architecture of the discriminator during the training process.

The discriminator also employs LeakyReLU activation in all its layers. As seen in equation (3) LeakyReLU activation function addresses the vanishing gradient problem by permitting a small, non-zero gradient for negative input values. It's a variation of the Rectified Linear Unit (ReLU) activation function and used to prevent the "dying ReLU" problem by allowing a small gradient even when the unit is not active. This helps maintain the flow of gradients during training.

$$Leaky\mathrm{Re}L\cup(x)=\begin{cases}x \quad x>0\\ 0.2x \quad x\leq 0\end{cases} \tag{3}$$

A regularization method called dropout is employed to avoid overfitting. It involves randomly deactivating a portion of neurons during the training process. Here, after each LeakyReLU activation, Dropout layers with a dropout rate of 0.25 are added.

The discriminator continues with a series of convolutional layers, gradually increasing the number of filters. These layers capture increasingly complex and abstract features in the input image. After the convolutional layers, a Flatten layer is added. In order to prepare the data for the fully connected layers, this layer flattens the 2D feature maps into a 1D vector. A single neuron connects the flattened vector to a dense (completely connected) layer. To represent the discriminator's confidence that the input image is real

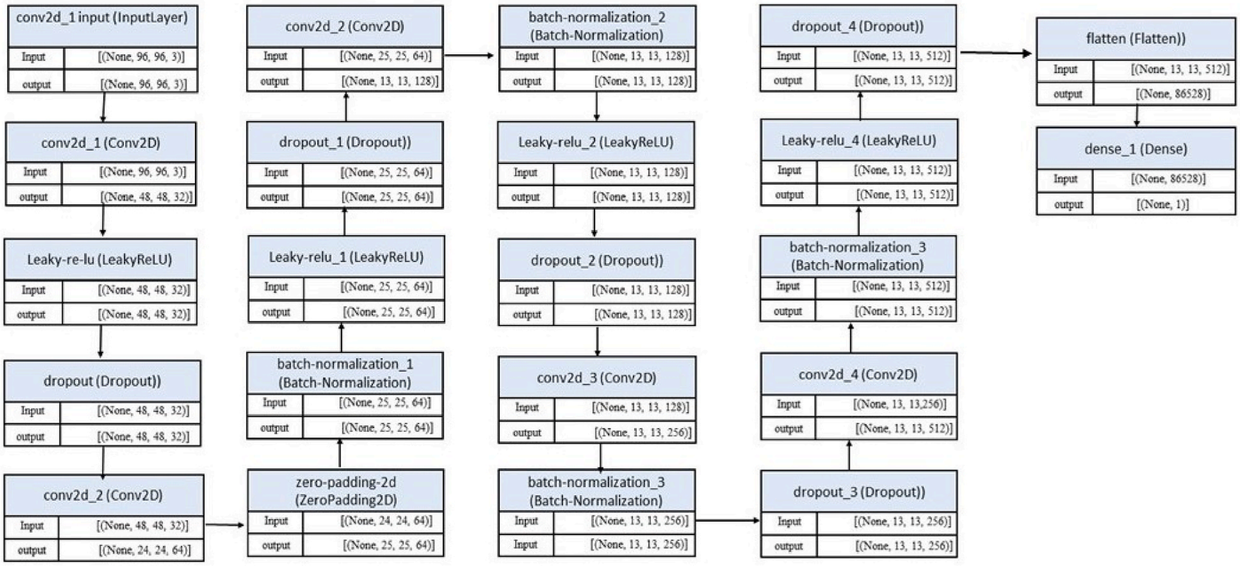

Fig. 7. Discriminator structure diagram.

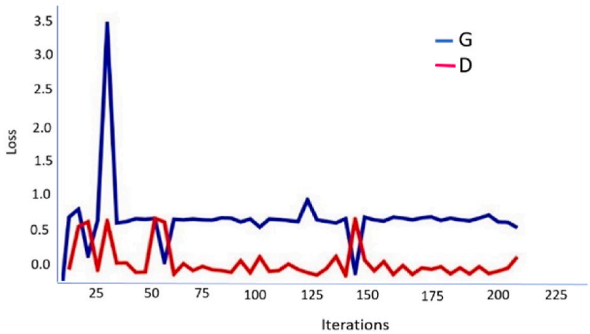

Fig. 8. Generator Discriminator loss during training.

(closer to 1) or fake (closer to 0), this layer employs the "sigmoid" activation function to produce an output between 0 and 1. Fig. 7 illustrates the structure of the discriminator in a visual form.

3.6. Training process of DCGAN

Training of DCGAN involves several steps.

The first step is to compute a loss function known as V(G). This loss function is designed to evaluate how well the generator (G) is performing in generating realistic data. The formula for V(G) considers a balancing parameter (y) and it measures the difference between the predicted data created by the generator (G(z)) and the actual data distribution.

$$v(G) = (1 - y)\log(1 - D(G(z))) \tag{4}$$

Equation (4) represents that 'z' is a 100-dimensional random noise vector that serves as input for the generator. 'G(z)' is the outcome or result produced by the generator when it takes in the noise vector 'z'. 'D(G(z))' represents the estimated probability that the discriminator assigns to the data produced by the generator.

In the next step, the DCGAN implementation involves calculating discriminator loss function called V(D). This loss function is used to assess the performance of the discriminator (D) in distinguishing between actual data and the data produced by the generator [36].

$$V(D) = -((1 - y)\log(1 - D((G(z))) + y\log D(x)) \tag{5}$$

In equation (5) *x* represents the real sample data obtained from the authentic dataset and *D(x)* represents the discriminator's estimated probability for this particular real data point.

In Fig. 8 graph shows the loss of both the generator and discriminator during the training process. The x-axis represents the training

Table 1
Parameters in neural network training.

Parameters	Value
Image scale	[-1, 1]
Optimizer	Adam
Learning rate	0.00001
Momentum rate	0.5
Batch size	32
Training iteration number	40
Dropout	0.25
Latent space dimension(Z)	100
Normalization	Batch normalization
Loss function	Binary cross-Entropy loss

Fig. 9. Results at epoch 40.

iterations, while the y-axis illustrates the corresponding loss values. In the initial phase, the generator's loss fluctuates from reasonable to high values. Concurrently, the loss for the discriminator exhibits small oscillations in response to real and fake samples.

The third step involves calculating the optimization function, which forms the core of the DCGAN model's training process. This mathematical representation is [33]:

$$\text{minmax}(D\ G) = E_{x-pdata(x)}[\log D(x)] + E_{x-p(x)}[\log(1 - D(G(z)))] \tag{6}$$

As described in equation (6) Pdata is real data distribution, Pz is noise distribution. D(x) near 1 for real input, D(z) near 0 for noise. Strong D: higher D(x), lower D(G(z)), larger V(D, G). Training ends when fake data resembles real and D can't differentiate. Table 1 comprises a list of parameters employed in the training of the neural network.

For assessing the quality of the stitching defect images produced by DCGAN, we utilize a comprehensive evaluation approach. This involves employing the human evaluation and Fréchet Inception Distance (FID) evaluation metrics. This evaluation strategy provides a thorough assessment of the performance of DCGAN in comparison to state-of-the-art methods. It assesses the quality of generated images produced by deep convolutional Generative Adversarial Networks. FID aims to measure how closely generated image distributions and real image distributions resemble each other, based on the statistics of feature vectors extracted from a pre-trained neural network. The Inception-v3 classifier, which is pre-trained network, used for feature extraction. We retrieve the embedding's of real and fake images after deleting the output layer in order to extract the feature distance.

4. Experiments

We trained DCGAN using a dataset of 6000 images containing uneven stitching, seam puckering and broken stitch defect categories. To enhance training efficiency and avoid loading all images into memory simultaneously, we employed a mini-batch training technique. Every 40 training batches, we generated a sample plot for local visual inspection. Through DCGAN, we generated different

Fig. 10. Results at epoch 80.

Fig. 11. Results at epoch 160.

samples with various types of defects. Fig. 9 shows the generated samples in different epoch cases.

Fig. 10 illustrates the generated images of unbalanced stitching defects at epoch 80.

Fig. 11 illustrates the generated images of unbalanced stitching defects at epoch 160.

Fig. 12. Generated image of unbalanced stitch using defective and non-defective fabric.

Fig. 13. Generated image of unbalanced stitch using pix2pix GAN.

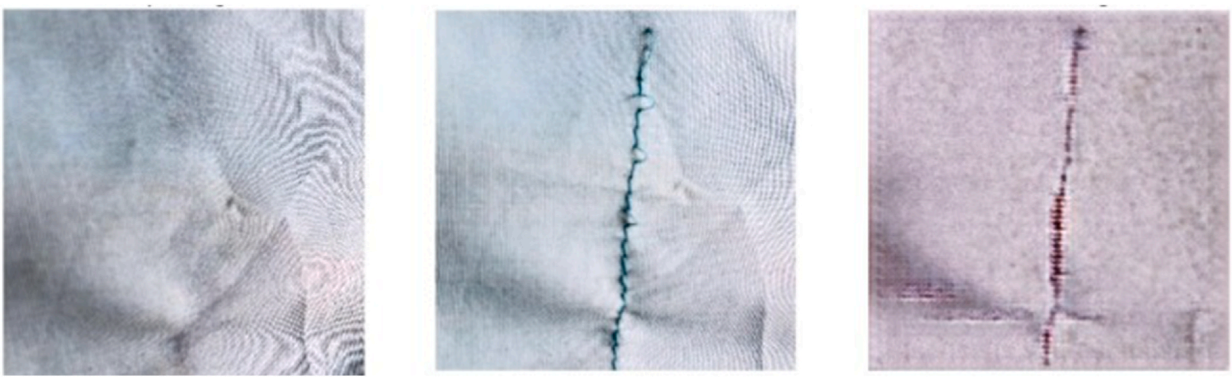

Fig. 14. Generated image of broken stitch using defective and non-defective fabric.

5. Experiments using pix2pix GAN

We also perform experiments using pix2pix GAN. pix2pix GAN, known for its image-to-image translation capabilities, excels in producing images that transform from one specific type to another. In stitching defects, it's effective at generating defective images of stitching by combining non defective patterns. However, this approach has a limitation It excels at creating variations of images based on those specific input pairs but lacks the capability to generate entirely new, unrelated images beyond the defined input pairs. In other words, it can't produce an infinite variety of stitching defects because it relies on the patterns and information present in the input images. In experiments, it combines defective and non-defective images to create variations within the boundaries of those pairs.

In contrast, DCGAN (Deep Convolutional Generative Adversarial Network) has the flexibility to generate entirely new images from random noise, offering a broader range of possibilities. Our proposed method not bound by the limitations of predefined input pairs

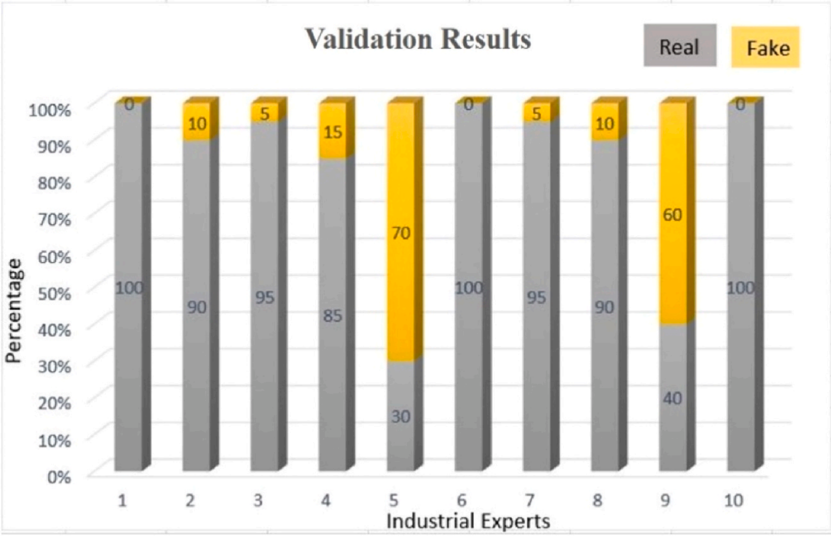

Fig. 15. Validation results of qualitative evaluation of generated stitching defects.

and can potentially generate a much larger and more diverse set of stitching defect images. It has the potential to create a much broader range of images of stitching defects because it's not limited by predefined input pairs.

Fig. 12 shows the outcomes of unbalanced stitch and broken stitch analysis, utilizing the Pix2Pix GAN model. The illustration involves taking both defective and non-defective fabric images and employing the Pix2Pix GAN to combine them, generating synthetic images that showcase stitching defects.

These Fig. 13 shows the outcomes of unbalanced stitch analysis, utilizing the Pix2Pix GAN model.

Fig. 14 displays the results of broken stitches using the Pix2Pix Generative Adversarial Network (GAN) model.

6. Evaluation

Evaluating generation models poses a considerable challenge. Human visual assessments or metric-based evaluations are the two main types of modern image quality evaluation methods. metric-based evaluation includes inception scores [34] Kernel Inception Distance (KID) and Fréchet Inception Distance (FID). In this article, the generated defect images are primarily intended for machine vision applications. Hence we also performed matrix based evaluation.

7. Qualitative evaluation

For validation of results, ten individuals from the stitching industry. We presented them with our generated results, showcasing images of apparel stitching defects. Remarkably, eight out of the ten participants were unable to differentiate between generated and actual images. They found the quality of the generated defects to be so convincing that it closely resembled actual stitching defects. However, two participants demonstrated the ability to identify the differences. As a result, our evaluation yielded an impressive accuracy rate of 85% percent. This outcome underscores the remarkable achievement of our generated images, as they convincingly replicated the characteristics of genuine stitching defects found in real-world scenarios.

Participant 1: These generated images look 100% real.

Participant 2: I would say they look 90% real.

Participant 3: I find them 95% real.

Participant 4: They appear about 85% real.

Participant 5: I can only see about 30% realism in them.

Participant 6: These look 100% real.

Participant 7: I'd say they are about 95% real.

Participant 8: I find them 90% real.

Participant 9: I can only see about 40% realism in these images.

Participant 10: These images look 100% real.

In Fig. 15 graph presents a visual representation of the assessment conducted by industrial experts. These outcomes provide valuable insights into the quality of the generated stitching defects.

Table 2
FID score during evaluation.

Types of Stitches	FID (Fréchet Inception Distance)
Unbalanced Stitch	61.43
Seam Puckering	65.74
Broken Stitch	75.55
Open Seam	69.54

Fig. 16. Original images of unbalanced stitching defect.

7.1. Quantitative evaluation

FID score for unbalance stitching is 61.43, for seam puckering 65.74 and for broken stitch 75.55. The FID (Fréchet Inception Distance) value serves as an indicator of the complexity of defects. A higher FID value signifies a greater level of complexity within the defect image [37]. Consequently, we can deduce that unbalance stitching defects are relatively simple, while broken stitch defects are the most complex type of defects.

Table 2 displays the FID metric, serving as an evaluation measure for assessing the quality of generated images related to unbalanced stitch seam puckering broken stitch and open seam.

In dataset generation of apparel stitching defects, the main focus on stitching details rather than the entire fabric. While evaluation metrics like Fréchet Inception Distance (FID) Inception score compare entire fake images with real ones, they may not capture the nuances of specific features like stitching lines effectively. This is because these matrices evaluate the overall image distribution and may not differentiate well if your primary concern is a small but crucial part of the image. In contrast, human evaluation allows you to prioritize the stitching aspects that matter most. By involving human experts from the stitching industry, one may get targeted feedback on the authenticity and quality of the stitching lines. This approach ensures that your generated dataset meets the specific requirements and standards of the stitching domain, even if it doesn't align perfectly with broader image-level metrics. In essence, human evaluation offers a more tailored and focused assessment that aligns better with objective of generating realistic stitching details rather than evaluating the entire fabric. It is to be mentioned that since this research does not involve experimental interventions on human subjects explicitly. However, the involvement of industrial experts in result validations implies that the study focuses on professional consultations or validations rather than experimental interventions on human participants.

8. Implementation details

Nvidia Tesla T4 GPU served as the training model environment for DCGAN having compute capability of 7.5 accelerator along with 2 x vCPU and 15360MiB memory (almost 12.68 GB useable), and it was executed on Google Colab's computing platform. Further, our methods and models are implemented in TensorFlow. The Adam optimizer is used to train both the generative network and the discriminative network, and a learning rate of 0.00001 is utilized. The training process uses a mini-batch size of 32. We save and restore model after every 40 epochs which is helpful in terms of long running training task. The iterations training are set as 40 100,

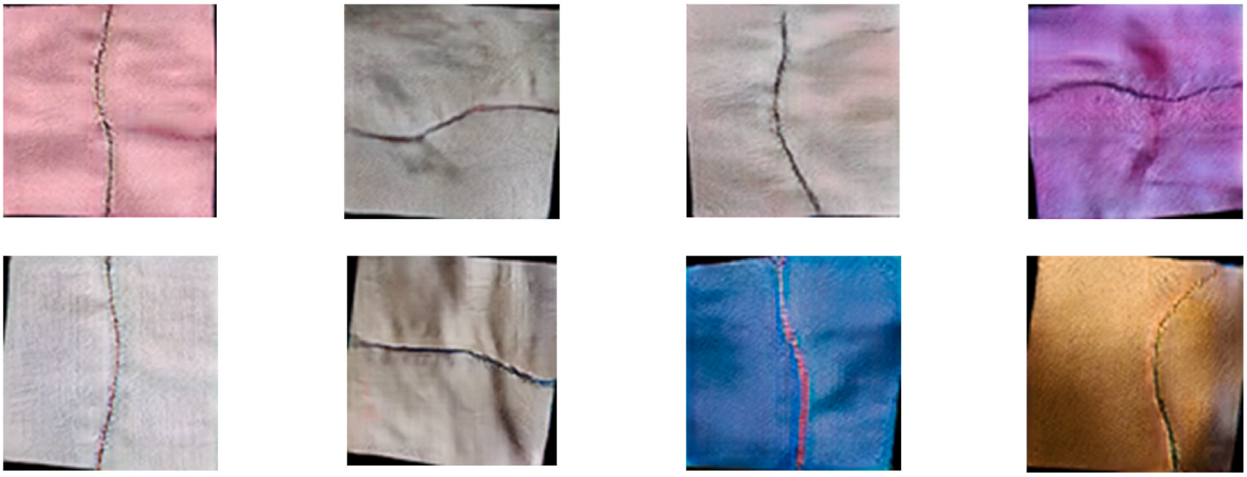

Fig. 17. Synthetic images of unbalanced stitching defects.

Fig. 18. Original images of seam puckering defects.

200, and 300 to 1000 iterations. After training, a total of 28 images are generated in every iteration. Moreover, the image sizes are 96, 96, 3 respectively, for output. Fig. 15 displays synthetic stitching defect images generated by DCGAN.

9. Image generation results

Original images of unbalanced stitch.

Fig. 16 illustrates the instances of unbalanced stitching defects from the original dataset.

Synthetic images of unbalanced stitch.

Fig. 17 displays synthetic images depicting unbalanced stitching defects that were generated using the DCGAN (Deep Convolutional Generative Adversarial Network) model.

Original images of seam puckering defect.

Fig. 18 illustrates the instances of seam puckering stitching defects from the original dataset.

Generated Images of seam puckering defect.

Fig. 19 showcases artificial images representing instances of seam puckering defects. Generated through the utilization of the

Fig. 19. Synthetic images of puckering seam defects.

Fig. 20. Generated images of broken stitches defect.

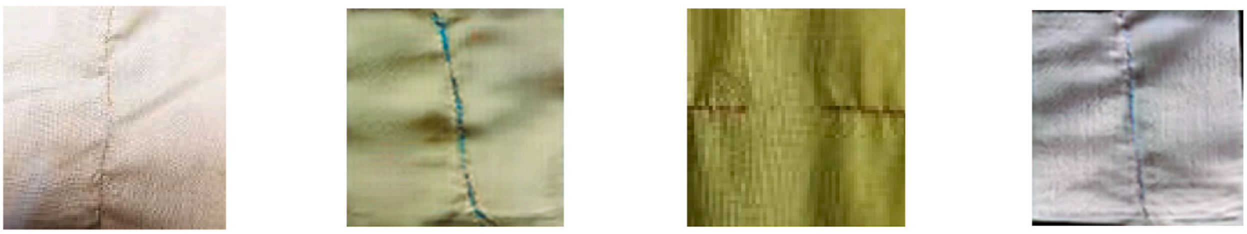

Fig. 21. Synthetic images of broken stitching defects.

Fig. 22. Original images of open seam defect.

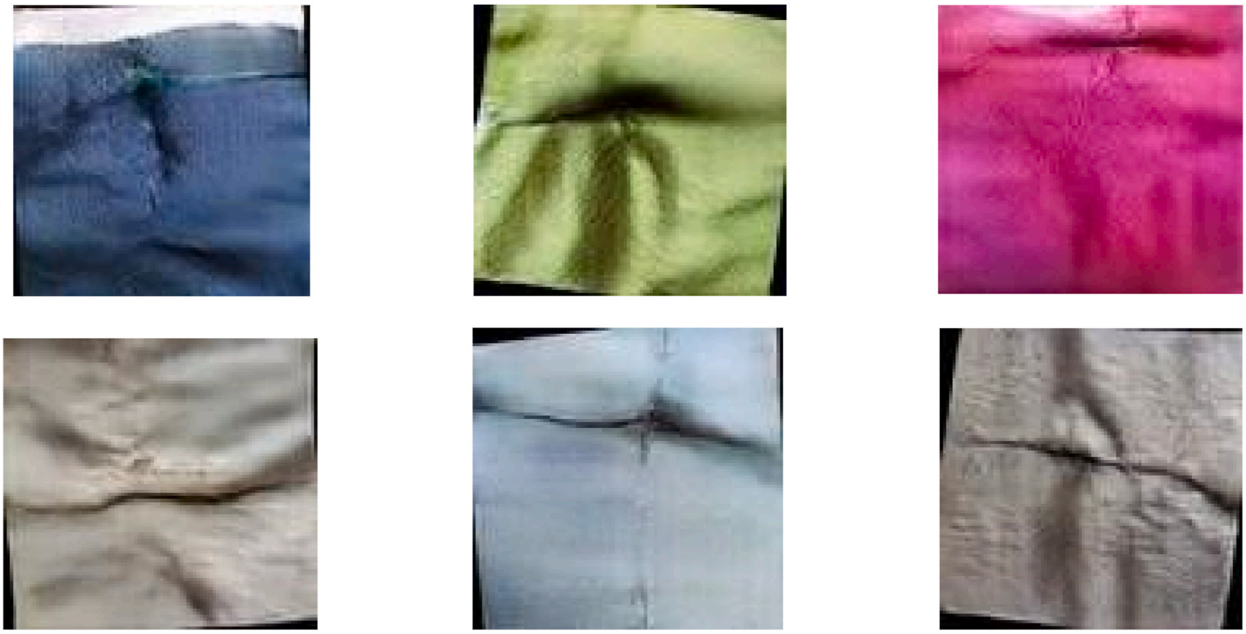

Fig. 23. Synthetic images of open seam defect.

DCGAN.

Generated images of broken stitch defect.

Fig. 20 illustrates the images of seam broken stitch defects from the original dataset.

Synthetic images of broken stitch defects.

Fig. 21 illustrates synthetic images of broken stitch defects generated through DCGAN.

Original images of open seam defect.

Fig. 22 shows the images of open seam defects from the real dataset.

Synthetic images of Open Seam defect.

Fig. 23 illustrates synthetic images of open seam defects generated through DCGAN.

10. Discussion and conclusion

The results of our experiments have shown that applying DCGAN to create stitching defects in an apparel dataset is viable and effective approach in practice. The data newly generated by the model remarkably contain the characteristics of the original data. This resemblance is so accurate that the discriminator, a key component in the GAN, struggles to differentiate between the generated and real data. When fake data of stitching defects added with original defect data it form a balance dataset.

In this paper, image generation network known as DC-GAN is applied. The primary objective is to address the challenge of having an insufficient dataset of stitching defect samples for use in defect detection tasks. DCGAN was developed to generate stitching defect images that exhibit both high quality and diversity. This work served to expand the dataset of stitching defects, simplifying the collection of such samples and helps in establishing a strong basis for the effectiveness of defect detection tasks specifically related to stitching defects. To validate the efficacy of proposed method for fabric defect generation, conducted a human evaluation involving industrial experts. This evaluation revealed an impressive accuracy rate of 80%, highlighting the excellent quality and realism of the fault images produced. Further, in the defect generation experiments, DCGAN outperformed by achieving lower Fréchet Inception Distance (FID) scores, demonstrating its superior performance in generating stitching defect images.

It's worth mentioning that this study primarily focused on a specific set of stitching defects. Future research endeavors could aim to broaden the scope by including a wider variety of stitching defects. The existing metrics used for evaluating neural network-generated data may not be comprehensive enough. Future scholars should consider expanding the range of evaluation metrics to assess generated data from a more comprehensive and multifaceted perspective.

CRediT authorship contribution statement

Noor ul-Huda: Investigation, Data curation. **Haseeb Ahmad:** Writing – review & editing, Validation, Supervision, Project administration, Conceptualization. **Ameen Banjar:** Resources, Funding acquisition. **Ahmed Omar Alzahrani:** Resources, Funding acquisition. **Ibrar Ahmad:** Methodology. **M. Salman Naeem:** Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R.A. Fayyaz, M. Maqbool, M. Hanif, Textile design generation using GANs, in: 2020 IEEE Canadian Conference on Electrical and Computer Engineering, CCECE), 2020, pp. 1–5.

[2] O. Rippel, M. Müller, D. Merhof, GAN-based defect synthesis for anomaly detection in fabrics, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA), 2020, pp. 534–540.

[3] X. He, Z. Luo, Q. Li, H. Chen, F. Li, "DG-GAN: a high quality defect image generation method for defect detection,", Sensors 23 (2023) 5922.

[4] L. Song, R. Li, S. Chen, Fabric defect detection based on membership degree of regions, IEEE Access 8 (2020) 48752–48760.

[5] W. Wang, N. Deng, B. Xin, Sequential detection of image defects for patterned fabrics, IEEE Access 8 (2020) 174751–174762.

[6] Y. Zhang, Z. Gao, C. Zhi, M. Chen, Y. Zhou, S. Wang, et al., A novel defect generation model based on two-stage GAN, E-Polymers 22 (2022) 793–802.

[7] Y. Li, D. Zhang, D.-J. Lee, Automatic fabric defect detection with a wide-and-compact network, Neurocomputing 329 (2019) 329–338.

[8] Y. Zhao, K. Hao, H. He, X. Tang, B. Wei, A visual long-short-term memory based integrated CNN model for fabric defect image classification, Neurocomputing 380 (2020) 259–270.

[9] M. Chen, L. Yu, C. Zhi, R. Sun, S. Zhu, Z. Gao, et al., Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization, Comput. Ind. 134 (2022) 103551.

[10] S. Wan, S. Goudos, Faster R-CNN for multi-class fruit detection using a robotic vision system, Comput. Network. 168 (2020) 107036.

[11] R. Ogawa, T. Kido, T. Mochizuki, Effect of augmented datasets on deep convolutional neural networks applied to chest radiographs, Clin. Radiol. 74 (2019) 697–701.

[12] A. Teramoto, T. Tsukamoto, Y. Kiriyama, H. Fujita, Automated classification of lung cancer types from cytological images using deep convolutional neural networks, BioMed Res. Int. 2017 (2017) 1–6.

[13] S. Meister, N. Möller, J. Stüve, R.M. Groves, Synthetic image data augmentation for fibre layup inspection processes: techniques to enhance the data set, J. Intell. Manuf. 32 (2021) 1767–1789.

[14] C. Dewi, R.-C. Chen, Y.-T. Liu, S.-K. Tai, Synthetic Data generation using DCGAN for improved traffic sign recognition, Neural Comput. Appl. 34 (2022) 21465–21480.

[15] Aug 13, Synthetic data generation using conditional-GAN, Available: https://towardsdatascience.com/synthetic-data-generation-using-conditional-gan-45f91542ec6b, 2021.

[16] P. Wang, Q. Chen, J. Li, L. Ma, M. Feng, Y. Han, et al., A microscopic traffic flow data generation method based on an improved DCGAN, Appl. Sci. 13 (2023) 7192.

[17] M. Mehralian, B. Karasfi, RDCGAN: unsupervised representation learning with regularized deep convolutional generative adversarial networks, in: 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium, 2018, pp. 31–38.

[18] K. Sun, Q. Wen, H. Zhou, "Ganster R-CNN: occluded object detection network based on generative adversarial nets and faster R-CNN,", IEEE Access 10 (2022) 105022–105030.

[19] S. Niu, B. Li, X. Wang, H. Lin, Defect image sample generation with GAN for improving defect recognition, IEEE Trans. Autom. Sci. Eng. 17 (2020) 1611–1622.

[20] D.-M. Tsai, C.-Y. Hsieh, Automated surface inspection for directional textures, Image Vis Comput. 18 (1999) 49–62.

[21] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[22] A. Van Den Oord, N. Kalchbrenner, K. Kavukcuoglu, Pixel recurrent neural networks, in: International Conference on Machine Learning, 2016, pp. 1747–1756.
[23] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, 2013 arXiv preprint arXiv:1312.6114.
[24] G. Zhang, Y. Pan, L. Zhang, Semi-supervised learning with GAN for automatic defect detection from images, Autom. ConStruct. 128 (2021) 103764.
[25] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415.
[26] Z.e. Abo-Hammour, O. Alsmadi, S. Momani, O. Abu Arqub, A genetic algorithm approach for prediction of linear dynamical systems, Math. Probl Eng. 2013 (2013) 1–12.
[27] Z. Abo-Hammour, O. Abu Arqub, S. Momani, N. Shawagfeh, Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm, Discrete Dynam Nat. Soc. 2014 (2014) 1–15.
[28] O. Abu Arqub, Z. Abo-Hammour, S. Momani, N. Shawagfeh, Solving singular two-point boundary value problems using continuous genetic algorithm, in: Abstract and Applied Analysis, 2012.
[29] G. Zhang, K. Cui, T.-Y. Hung, S. Lu, Defect-GAN: high-fidelity defect synthesis for automated defect inspection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 2524–2534.
[30] J. Ma, W. Yu, P. Liang, C. Li, J. Jiang, FusionGAN: a generative adversarial network for infrared and visible image fusion, Inf. Fusion 48 (2019) 11–26.
[31] Y. Wang, B. Yu, L. Wang, C. Zu, D.S. Lalush, W. Lin, et al., 3D conditional generative adversarial networks for high-quality PET image estimation at low dose, Neuroimage 174 (2018) 550–562.
[32] G. Hu, J. Huang, Q. Wang, J. Li, Z. Xu, X. Huang, Unsupervised fabric defect detection based on a deep convolutional generative adversarial network, Textil. Res. J. 90 (2020) 247–270.
[33] J. Liu, C. Wang, H. Su, B. Du, D. Tao, Multistage GAN for fabric defect detection, IEEE Trans. Image Process. 29 (2019) 3388–3400.
[34] B. Li, X. Qi, P. Torr, T. Lukasiewicz, Lightweight generative adversarial networks for text-guided image manipulation, Adv. Neural Inf. Process. Syst. 33 (2020) 22020–22031.
[35] Y. Du, W. Zhang, J. Wang, H. Wu, "DCGAN based data generation for process monitoring,", in: 2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS), 2019, pp. 410–415.
[36] H. Wang, M. Gao, S. Hu, Z. Sun, Z. Xu, Study on weather radar echo data generation based on DCGAN, IEEE Access 7 (2019) 131978–131985.
[37] A. Borji, Pros and cons of GAN evaluation measures: new developments, Comput. Vis. Image Understand. 215 (2022) 103329.