Industrial Product Defect Detection Using Custom U-Net

Al Amin
Energy and Electronic Engineering
University of portsmouth
Portsmouth, United Kingdom
0000-0002-2059-2352

Dr. Hongjie Ma
Senior Research Fellow
University of Portsmouth
Portsmouth, United Kingdom
0000-0001-8507-3636

Md. Shazzad Hossain
Research and Development
Time Research and Innovation
Dhaka, Bangladesh
0000-0001-9507-5395

Nasim Ahmed Roni
Research and Development
Time Research and Innovation
Dhaka, Bangladesh
nasimahmedroni@gmail.com

Erfanul Haque
Research and Development
Time Research and Innovation
Dhaka, Bangladesh
ehaquedipto@gmail.com

S M Asaduzzaman
Research and Development
Time Research and Innovation
Dhaka, Bangladesh
0000-0001-7058-2606

Redwan Abedin
Research and Development
Time Research and Innovation
Dhaka, Bangladesh
redwanabedin@gmail.com

Alif B Ekram
Research and Development
Time Research and Innovation
Dhaka, Bangladesh
alifbekram@gmail.com

Rube Farzana Akter
Energy and Electronic Engineering
University of Portsmouth
Portsmouth, United Kingdom
0000-0003-0314-8329

***Abstract*—Numerous automated labour-saving systems have been created and implemented to lower production costs and enhance product quality. Systems for intelligent visual analysis are now playing a more significant role in production lines. Many deep learning and machine learning techniques were previously used to identify defective products. Still, the models were never tested because they did not produce satisfactory results and had numerous other problems adjusting to sparse and poor-quality data. To address this issue, a deep learning-based customised U-Net model was introduced in this proposed work. This model was trained on six different classes of data-sets to assess the model's capability on different image textures and resolutions. The proposed model had an overall accuracy of 97.25%. We also achieved significantly h igher p recision, r ecall, a nd F 1 s cores of 96.04%, 95.58%, and 96.12%, respectively. The execution times of six different classes can indicate how well a model performs, and the average execution time is only 33.1 seconds, according to observations.**

***Index Terms*—defect detection, optimised U-Net, industrial production model, U-Net 3+, convolutional auto-encoder**

I. Introduction

The quality of manufactured products is easily affected in the industry due to deficiencies and limitations of the existing techniques, working conditions, and other factors. Product quality is the primary concern in industrial production. In mass production, it is crucial to check the quality of the finished goods [1]. Manual inspection is the gold standard for identifying surface flaws i n m anufactured g oods. M anual defect inspections have a low accuracy rate because of subjective factors and the inspector's work experience. Neither option is viable when a flaw is too small to be seen by the naked eye and too dangerous to inspect by hand. Visual defect detection has recently replaced manual defect detection as a standard method for checking the exterior of manufactured goods. Surface defect detection on industrial products is an automatic, hands-free detection method that uses visual perception technology. It has a high level of accuracy and can function for long periods in a demanding manufacturing setting. Defects in ceramic tiles [2], textiles [3], steel plates [4], and printed circuit boards [5] are just some of the common uses.

It is possible to enhance production efficiency greatly and cut labour costs by inspecting industrial products for obvious surface flaws. Using one's eyes alone, there are primarily two methods for detecting surface defects in industrial products: the first uses traditional image processing, while the second uses deep learning. Image processing has been used for a very long time to identify surface defects on an industrial product manually. The manual design of features does not require a large amount of data for feature learning. Still, it is less adaptable and has more stringent imaging environment requirements. Deep learning-based surface defect detection for industrial products requires the automated extraction of features from massive data sets. Despite its flexibility, this automatic feature extraction method necessitates a large amount of training data. Since flawed samples are unlikely to be encountered in a real-world industrial setting, collecting them and correctly labelling them will be time- and labour-intensive. Consequently, the ability to detect surface defects in industrial products using a minimal amount of labelled data is crucial.

This article details a method for identifying surface defects

in manufactured goods using only a minimal amount of labelled data. A large portion of the process is based on either conventional image processing or deep learning techniques that rely on a limited amount of labelled data to detect flaws in the surfaces of industrial products. We present the following outline for this paper. Although researchers presented different machine learning and deep learning-based techniques, some limitations, like the dataset, must be appropriately addressed.

The main contributions of this paper can be summarized as follows:

1) This research suggested a compact convolutional neural network (CNN)-based tailored technique termed U-Net architecture to identify a few frequent fabric defects.
2) Our modified U-Net model adds more skip connections between the various decoder and encoder levels than the conventional U-Net structure.
3) Before each decoding stage, we additionally added a concatenate layer so that the outcomes of each encoder stage and the previous encoding stage could be combined.
4) The suggested network outperformed conventional CNN architectures regarding detection accuracy using a substantially smaller model size.
5) With much superior precision, recall, and F1 scores of 96.04%, 95.58%, and 96.12%, the proposed model was trained on six different data sets classes to test its performance on various image textures and resolutions.
6) The success of the proposed network in detecting fabric flaws and executing six different types of tasks demonstrates how well the model works compared to previous models.

This study uses a custom model to detect defective industrial products. As industries struggle with accumulating defective products, this issue was chosen to solve the data set's shortcomings with an efficient p erformance. T he r emainder o f the work is organised in this manner to address the difficulty of the small dataset. The earlier investigation on defect detection is presented in Section 2. The methodology, comprising the model diagram, dataset description, data preparation, and data modelling, is illustrated in Section 3. The result analysis section, which includes result comparison, visualisation, and data validation, is shown in Section 4. The comprehensive explanation of this suggested system is covered in Section 5, and the conclusion and future work are covered in Section 6.

II. Related Works

The fabric has numerous domestic and industrial applications, including clothing, blankets, sheets, air suits, filter cloth, and other similar garments and accessories [6]. Any company's reputation is directly proportional to the calibre of its products. If a company produces defective products, it will waste not only raw materials but also lose many customers. In addition, the life-saving properties of particular medical and space fabrics may be correlated with the effectiveness of spacecraft stabilisers. Fabric inspection, as a mechanism for quality control, is essential for ensuring textile quality. Most fabric defect inspections are conducted visually by human workers due to high labour costs and high demand for skilled labourers [7].

Numerous computer vision techniques have been created and applied to the problem of textile defect detection. The primary types of defect recognition methods are model-based, filter-based, learning-based, and feature-based methods [8]. Jing and others on the plain fabric were able to achieve satisfactory results, but on patterned fabric, it was impossible to segment the defects precisely [9]. Due to the repeating texture structure of the fabric, Li et al. proposed a low-rank representation technique for defect detection [10]. Dimension reduction via eigenvalue decomposition is laborious, so processing block images requires more time. Using the saliency metric for colour dissimilarity and positional aggregation, it was also proposed that a novel method for identifying fabric defects could be achieved [11]. Due to the numerous similarity calculations performed by the algorithm, the time complexity needs to be enhanced. However, the algorithm detects patterned fabric effectively. Zhang et al. used L0 gradient minimisation and C-means clustering to eliminate the fabric's background texture and segment the defects, respectively [12]. Even though this method can detect flaws in fabrics with a wide variety of textures, it requires manual adjustment of several parameters. In their study, Kang et al. demonstrated a method for defect segmentation in textiles using a single, uncomplicated image and an Elo-rating algorithm [13]. Using this method, fabrics with a short period texture can be easily segmented, whereas those with an extensive period texture cannot be segmented. According to Guan et al., task-specific characteristics are typically crafted by hand using the above-mentioned techniques [14].

Convolutional neural networks (CNNs) have grown in popularity in recent years for tasks including image classification, object detection, and image segmentation [15]. In contrast to conventional image processing techniques, CNN can automatically learn relevant features from data without the need for intricately designed features [16]. In light of the success of deep learning techniques in other application areas, they are now being applied to detect fabric defects [17]. Defect detection can be divided into four stages, from coarsest to finest. Classification, localisation, segmentation, and semantic segmentation are the stages involved. Specification of defect detection entails sorting out the flaws present and examining the affected area to determine its exact location and nature. The primary objective of defect segmentation is to determine whether or not each pixel in a fabric image represents a defect. In addition to determining whether or not each pixel contains defects, semantic defect segmentation classifies the defects.

Li et al. developed a compact CNN architecture to identify and classify a handful of prevalent fabric defect detection [18]. The proposed network demonstrated superior detection accuracy despite employing a significantly smaller model size than conventional CNN architectures. Utilising the exceptional feature extraction capabilities of AlexNet, a custom classification system for flaws in yarn-dyed fabrics was developed

[19]. Wei et al. combined CNN and compressed sensing for defect classification due to CNN's ineffectiveness in dealing with the problem of small sample sizes in classification tasks [20]. To expedite detection, a YOLO-based defect localisation algorithm has been proposed [21]. In experimental settings, the network successfully identified five types of fabric defects. Li et al. proposed a network for defect localisation that relies on identifying surface flaws to pinpoint the exact location of a problem [22]. Prior research focused primarily on identifying and classifying defects.

In contrast, the class imbalance is prevalent in real-world applications, making it difficult to extract useful CNN model features or even causing the model to over-fit [23]. This issue affects numerous practical applications, such as fraud detection, anomaly detection, facial recognition, etc. [24]. Two methods for addressing class imbalance in deep learning are hard sampling and soft sampling. Class imbalance can be corrected by down-sampling positive or negative samples using hard sampling methods. Jian et al. proposed a sampling-based method to address the issue of sample imbalance in detecting mobile phone screen defects [25]. Soft sampling techniques typically generate a weighted loss function, increasing the minority sample weighting. Soft sampling strategies do not involve sample discarding; the entire dataset is utilised to fine-tune the parameters.

The previous background investigation revealed that the other researchers used techniques related to automated product defect identification with varying degrees of success. It evaluated their work and identified a variety of issues, such as the failure to recognise defects in dis-patterned fabrics [8], the use of low-ranked representation techniques for defect detection [11], [12], the manual adjustment of several feature extraction parameters [12], the presence of insufficient negative samples that reduce detection accuracy [26], and the inability to see real-time flaws in automatic defect detection [23]. Over fitting issues with several models were also discovered [24], [25].

So regarding these, our model uses the image reconstruction technique, in which only non-faulty data is provided to the model during training and uses any "out of distribution" data to identify the defective data. The benefits of the method we recommend for identifying fabric flaws are: Our customised 2D U-Net is a ground-breaking technique that employs only non-defective data to discover the faults, finally reducing the algorithm's complexity. This is in contrast to previous defect identification and segmentation approaches. Utilising this method lowers the cost of obtaining labelled defect data. Additionally, very little data is used to accomplish the purpose. Furthermore, this particular method can detect flaws in both patterns.

III. Methodology

The suggested method for the study is made up of four main parts: the data set, the pre-processing of the data, training and testing, and the development, testing, and evaluation of the model. In our analysis, we used six data classes from the DAGM2007 dataset. These data classes included defect and non-defect classes of different industrial goods. After that, the data was handled in a number of ways, such as by slicing, rotating, and resizing. After building the model, training it, and evaluating it, the data was split into two groups: training and testing. Finally, it was determined how well the model performed after the training phase was completed.

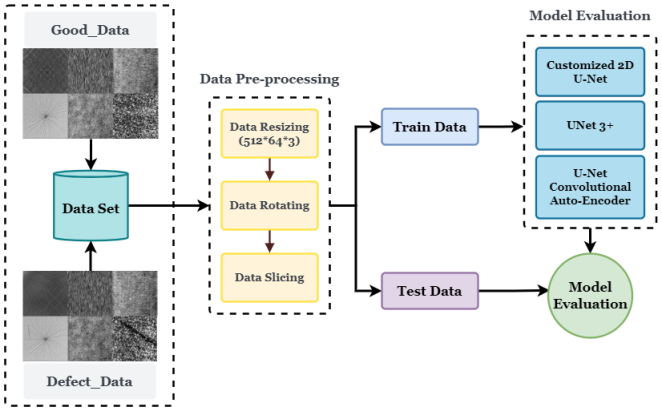


Fig. 1: Overall Pipeline for Defect Detection

A. Dataset

This research utilized the DAGM 2007 Competition Dataset [27]. The data is generated purposefully and is analogous to real-world issues. It consists of six data class sets, each of which has 1,000 images of a background texture with no defects and 151 with a single flaw annotation. With ellipses, we indicate weak labeling. A grayscale, 8-bit PNG picture is used to improve defective images. 0 and 255 indicate the background and error areas, respectively. It is true that the data is unbalanced and there are fewer records in the fault class, which could affect the performance of the model. In real-world industrialization, the flaw happens less often, so the model should be able to find it easily.

B. Data Pre-processing and Test Train Split

Data pre-processing is a crucial part of any Deep Learning model. This project also has the pre-processing steps for the image dataset. The problem of data imbalance between the defect and non-defect classes is first addressed by using an up-sampling strategy for the defect class to generate more artificial images. The images are in (512*512*3) pixels, but before going to the training phase, we resize the images in (512*64*3) pixels. Also, the image was cropped and rotated to look better into the affected areas. Lastly, the image is sliced into 30 pieces to closely look at the affected areas taken into the image slices. To train the network, a NumPy array from stored sliced images was made to prevent bias in the training data, and all the images were randomly shuffled. Using the split techniques, libraries allocate 20% of the photos for testing and 80% of the data for training at random. Therefore, for each class, we use 921 images for training and 230 for testing.

C. Proposed Customised U-Net Model Building

The proposed customized U-Net has caused significant modifications to the U-Net model. Our custom U-Net model was made by combining the existing U-Net design with a few extra skip connections between the different tiers of the decoder part and the encoder part. Also, we added a concatenate layer at the beginning of each decoding level to connect the outputs from different encoder levels and the output from the level before it i Fig 2. If information is lost during the downsampling, it can be sent straight to the decoder.

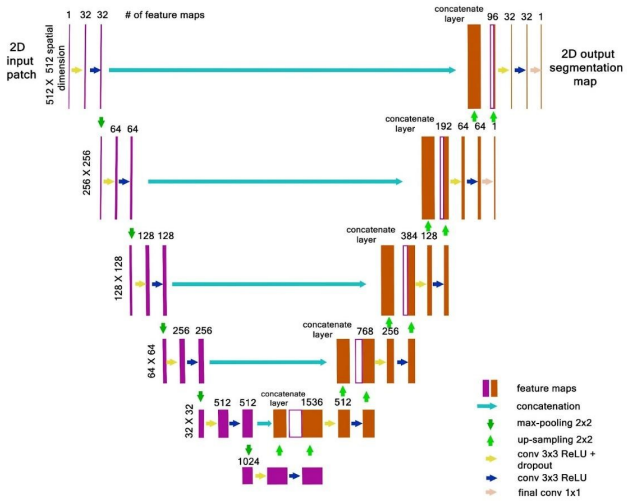


Fig. 2: Proposed Custom U-Net Architecture

Result Analysis

Three different models were used to examine the performance of six different types of data classes in this experiment. When assessing each model's output, accuracy, precision, recall, F1 score, and the overall process execution time are all considered. Table 1 shows that the proposed custom U-Net model performed better in six different classes (class 1- 98.03%, class 2- 97.75%, class 3- 94.53%, class 4- 98.52%, and class 5- 95.94%, class 6- 98.76%). In contrast, the Convolutional Auto-encoder performed better than U-Net 3+ in most cases except one (class 3), where Convolutional Auto-encoder achieved 91.43%, and U-Net 3+ got 93.60

This also demonstrated that custom U-net outperforms standard U-net in terms of F1 score (96.12%), recall (95.58%), and precision (96.045%). The U-Net 3+ model is found to have the lowest overall precision, recall, and F1 score value. The convolutional auto-encoder's recall is slightly lower than the average recall for the customised U-Net Model, which is marginally higher at 95.5% and 95%, respectively. The average recall rate for U-Net 3+ is 90.7%.

The most noteworthy finding i s t hat t his c ustomised U-net completes the classification process for all six classes faster than the Convolutional encoder, which came in second, and U-Net3+, which came in last. In this study, it is evident that the customised U-net performs better. Therefore, it can be concluded that in all of the cases, the proposed custom U-Net architecture maintained the most promising results, followed by Convolutional Auto-encoder in second place and U-Net 3+ in third.

D. Confusion Matrix

The confusion matrices used in the class 1 model evaluation are shown in Fig. 3. We will display three confusion matrices for each model. Since the Custom U-Net model performed the best in this category, the confusion matrix of class-4 data for all three models was shown in this instance. The results of the other two models are inferior to those of the proposed model, even though the same procedure was applied.

The customised model correctly predicted all but one of the images while detecting class-4 defects. Although the image was flawed, the model failed to spot it. Convolutional Auto-encoder experienced a similar situation, except that it reported a false positive for two images in the data-set's "Not-Defective" class. However, U-Net 3+ had two false positive images in both the "Defective" and "Not Defective" sections.

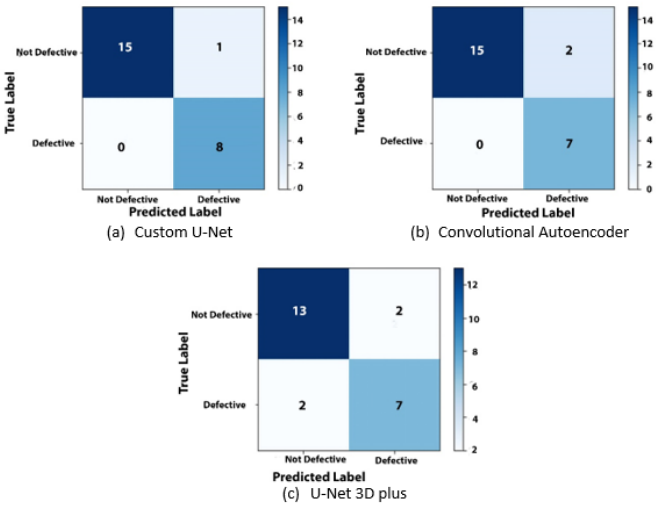


Fig. 3: Confusion Matrix of 3 Different Models

E. Loss and Accuracy

study, accuracy and Loss sources are represented by essential and useful graphs in Fig. 4. The confusion matrix was displayed using the same methodology for the accuracy and loss values. The proposed models showed almost no fluctuation during the training period for both training and reasonable accuracy.

Only our Customised Model provided a smooth curve during training, despite the Loss. Convolutional Auto-encoder loss and U-Net 3+ Loss experienced a slight uptick in performance during the first training session. After that, they both gradually started to decline.

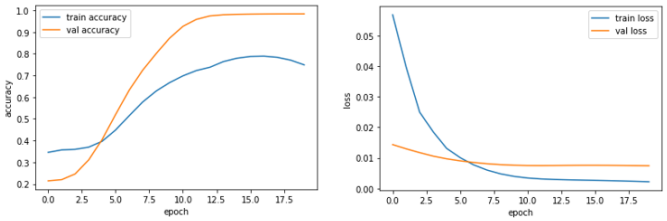
TABLE I: EXPERIMENTAL RESULTS FOR DIFFERENT MODELS

Models	**Data Class**	**Accuracy**	**Precision**	**Average**	**Recall**	**Average Recall**	**F1 Score**	**Average F1**	**Executing time (s)**
Proposed model	1	0.9803	0.9724	0.96045	0.9801	0.955875	0.9733	0.961266667	29.8
	2	0.9775	0.972		0.9782		0.9653		29.5
	3	0.9453	0.9251		0.9135		0.9235		31.8
	4	0.9852	0.974		0.96875		0.9843		31.3
	5	0.9594	0.9543		0.9489		0.9591		44.7
	6	0.9876	0.9649		0.9458		0.9621		31.5
Convolutional Autoencoder	1	0.9732	0.9725	0.951033333	0.9731	0.950233333	0.9686	0.953333333	44.4
	2	0.9564	0.9542		0.9487		0.9524		30.4
	3	0.9143	0.9126		0.9057		0.9105		34.8
	4	0.9627	0.9588		0.9583		0.961		30.1
	5	0.9467	0.9358		0.9423		0.9455		47.5
	6	0.9868	0.9723		0.9733		0.982		37.7
U-Net 3+	1	0.9236	0.9235	0.907933333	0.9226	0.915383333	0.9214	0.9168	83
	2	0.9287	0.9204		0.9135		0.9258		64
	3	0.936	0.9324		0.9431		0.9258		66
	4	0.9139	0.9056		0.9241		0.9125		66
	5	0.9206	0.9124		0.9028		0.9203		84
	6	0.8952	0.8533		0.8862		0.895		65

TABLE II: COMPARISON WITH PAST STUDIES

Sl No	Author	Model	Accuracy
1	Lin et al. [28]	Class Activation Map Guided U-Net (CAM-UNet)	72%
2	Yu-Jie et [29]	Attention U-Net with Feature Fusion Module	95.6%
3	Vivian et al. [30]	3D U-Net	88.4%

It is also evident from Fig. 4 that none of the models displayed an overfitting issue as training came to an end, with accuracy and Loss during exercise and validation being nearly equal.



a. Custom U-Net Accuracy b. Custom U-Net Loss

Fig. 4: Accuracy & Loss Graph of Custom U-Net Model

IV. Discussion

Suppose we focus on an image going through a down-sample stage. In that case, it loses information primarily related to the high-frequency components of the image, which can help us understand the workings of the proposed method. Down-sampling would almost always occur on an over-sampled signal in a system or be followed by up-sampling for rate conversion. The data points affected in both situations were already needed for processing. Therefore, it's always beneficial t o r emember w hat t ook p lace b efore t he down-sampling.

However, it also reduces the likelihood of accuracy loss in classification, so in that situation, some additional skip layers are used here to keep that information. The extra concatenate layer takes inputs and concatenates them along a predetermined dimension, giving the convolutional layers the advantage of remembering every lost detail for clarification. They lessen the problem of vanishing gradients by allowing gradients to flow through the model more freely. They enable features that could be lost due to down-sampling on the encoder side of the network to be added from the encoder side to the decoder side of the network. Because the left two models in this experiment lack the concatenate mechanism, the proposed customised U-Net model performed better than another model.

In addition, our suggested model is superior to the other model utilized previously in this defect detection field, as shown in Table 2, which lists three earlier works, none of which produced better performance than this proposed custom U-Net model.

Although this proposed architecture performed well in various types of data classes, it does not imply that it would perform better with any new challenges. The critical problem is determining that when an industrial-based defect detection situation occurs, there will be many challenges, such as when the data amount is so tiny that the model cannot cope with it. As a result, it can be regarded as a potential future limitation of the proposed mechanism.

V. Conclusion and Future Direction

This research created a deep learning-based, specialised U-Net model for identifying industrial product problems using recent advancements in convolutional neural networks and the U-Net architecture. Six different synthetic data-sets that were

presented at the DAGM 2007 Symposium were used to assess the approach that has been proposed. To identify the defects in fabrics and distinguish between bad and good fabrics, we used three models in this work as classifiers: a customised U-Net, a U-Net Convolutional Auto Encoder, and U-Net 3+. The U-Net model has undergone significant modification due to the proposed customised U-Net. The traditional U-Net architecture was changed to include a few more skip connections between the various layers of the decoder component and the encoder portion to create our customised U-Net model.

Additionally, we added a concatenate layer at the start of every decoding level so that the outputs from the various encoder levels and the level of encoding that came before it could be combined. The investigation results showed the suggested customised U-Net model performed significantly better than a different model and obtained an average accuracy of 97.25%. We also received substantially better precision, recall, and F1 scores of 96.04%, 95.58%, and 96.12%. Observations show that the execution times of six different classes can indicate how well a model performs and that the average execution time is only 33.1 seconds (s). By using several pre-processing methods, such as resizing, rotating, and slicing the data, the performance of the customised U-Net model was greatly enhanced.

Moreover, working with industrial data is challenging since it is inadequate, and imaginative work requires employing comprehensive data and pre-processing terms to extract the pertinent aspects. This is a contribution based on the idea of building a gateway with data pre-processing techniques that can identify faulty images when teaching models with varied dimensional parameters using a customised U-Net method instead of a hybrid approach. Although this suggested architecture worked well across a range of data classes, this does not necessarily mean it would perform even better under new conditions. The crucial issue is identifying the numerous difficulties that may arise in an industrial-based defect detection scenario, such as when the data amount is so tiny that the model cannot handle it. It can therefore be seen as a potential future limitation of the suggested mechanism.

REFERENCES

[1] M. Bradshaw, "The application of machine vision to the automated inspection of knitted fabrics," *Mechatronics*, vol. 5, no. 2-3, pp. 233–243, 1995.

[2] R. Stojanovic, P. Mitropulos, C. Koulamas, Y. Karayiannis, S. Koubias, and G. Papadopoulos, "Real-time vision-based system for textile fabric inspection," *Real-Time Imaging*, vol. 7, no. 6, pp. 507–518, 2001.

[3] C. Ciamberlini, F. Francini, G. Longobardi, P. Poggi, P. Sansoni, and B. Tiribilli, "Weaving defect detection by fourier imaging," in *Vision Systems: Applications*, vol. 2786, pp. 9–18, SPIE, 1996.

[4] Y. Zhiwei, D. Ming, and Z. Zhiheng, "Surface defect detection method of industrial products based on histogram difference[j]," *Graphic and image processing in general*, vol. 47, no. z1, pp. 247–249,267, 2020.

[5] L. Rao and J. Junfeng, "Classification method of fabric surface defects based on convolution neural network[j]," *Measurement amp; Control Technology*, vol. 9, p. 20–25, 2018.

[6] D. Yapi, M. S. Allili, and N. Baaziz, "Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1014–1026, 2017.

[7] Y. Wu, J. Zhou, N. T. Akankwasa, K. Wang, and J. Wang, "Fabric texture representation using the stable learned discrete cosine transform dictionary," *Textile Research Journal*, vol. 89, no. 3, pp. 294–310, 2019.

[8] B. Mallik-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Textile Research Journal*, vol. 70, no. 9, pp. 758–762, 2000.

[9] J. Jing, P. Yang, P. Li, and X. Kang, "Supervised defect detection on textile fabrics via optimal gabor filter," *Journal of Industrial Textiles*, vol. 44, no. 1, pp. 40–57, 2014.

[10] P. Li, J. Liang, X. Shen, M. Zhao, and L. Sui, "Textile fabric defect detection based on low-rank representation," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 99–124, 2019.

[11] K. Zhang, Y. Yan, P. Li, J. Jing, X. Liu, and Z. Wang, "Fabric defect detection using salience metric for color dissimilarity and positional aggregation," *IEEE Access*, vol. 6, pp. 49170–49181, 2018.

[12] H. Zhang, J. Ma, J. Jing, and P. Li, "Fabric defect detection using l0 gradient minimization and fuzzy c-means," *Applied Sciences*, vol. 9, no. 17, p. 3506, 2019.

[13] X. Kang and E. Zhang, "A universal defect detection approach for various types of fabrics based on the elo-rating algorithm of the integral image," *Textile Research Journal*, vol. 89, no. 21-22, pp. 4766–4793, 2019.

[14] S. Guan and H. Shi, "Fabric defect detection based on the saliency map construction of target-driven feature," *The Journal of the Textile Institute*, vol. 109, no. 9, pp. 1133–1142, 2018.

[15] Y. Lu, "Artificial intelligence: a survey on evolution, models, applications and future trends," *Journal of Management Analytics*, vol. 6, no. 1, pp. 1–29, 2019.

[16] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020.

[17] R. M. Alguliyev, R. M. Aliguliyev, and F. J. Abdullayeva, "Privacy-preserving deep learning algorithm for big personal data analysis," *Journal of Industrial Information Integration*, vol. 15, pp. 1–14, 2019.

[18] Y. Li, D. Zhang, and D.-J. Lee, "Automatic fabric defect detection with a wide-and-compact network," *Neurocomputing*, vol. 329, pp. 329–338, 2019.

[19] J. Jing, A. Dong, P. Li, and K. Zhang, "Yarn-dyed fabric defect classification based on convolutional neural network," *Optical Engineering*, vol. 56, no. 9, p. 093104, 2017.

[20] B. Wei, K. Hao, X.-s. Tang, and Y. Ding, "A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes," *Textile Research Journal*, vol. 89, no. 17, pp. 3539–3555, 2019.

[21] H.-w. Zhang, L.-j. Zhang, P.-f. Li, and D. Gu, "Yarn-dyed fabric defect detection with yolov2 based on deep convolution neural networks," in *2018 IEEE 7th data driven control and learning systems conference (DDCLS)*, pp. 170–174, IEEE, 2018.

[22] Y. Li, H. Huang, Q. Xie, L. Yao, and Q. Chen, "Research on a surface defect detection algorithm based on mobilenet-ssd," *Applied Sciences*, vol. 8, no. 9, p. 1678, 2018.

[23] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[24] A. Abdallah, M. A. Maarof, and A. Zainal, "Fraud detection system: A survey," *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.

[25] C. Jian, J. Gao, and Y. Ao, "Imbalanced defect classification for mobile phone screen glass using multifractal features and a new sampling method," *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 24413–24434, 2017.

[26] Y. Li, W. Zhao, and J. Pan, "Deformable patterned fabric defect detection with fisher criterion-based deep learning," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1256–1264, 2016.

[27] M. H. Skjelvareid, "Dagm 2007 competition dataset," Jun 2019.

[28] Y.-J. Xiong, Y.-B. Gao, H. Wu, and Y. Yao, "Attention u-net with feature fusion module for robust defect detection," *Journal of Circuits, Systems and Computers*, vol. 30, no. 15, p. 2150272, 2021.

[29] Y.-J. Xiong, Y.-B. Gao, H. Wu, and Y. Yao, "Attention u-net with feature fusion module for robust defect detection," *Journal of Circuits, Systems and Computers*, vol. 30, no. 15, p. 2150272, 2021.

[30] V. W. H. Wong, M. Ferguson, K. H. Law, Y.-T. T. Lee, and P. Witherell, "Automatic volumetric segmentation of additive manufacturing defects with 3d u-net," *arXiv preprint arXiv:2101.08993*, 2021.