IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 May 2025, accepted 19 July 2025, date of publication 28 July 2025, date of current version 13 August 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3593335

RESEARCH ARTICLE

Real-Time Defect Detection for Fast-Moving Fabrics on Circular Knitting Machine Under Various Illumination Conditions

YAN-QIN NI[1], PEI-KAI HUANG[2], (Graduate Student Member, IEEE), CHING-HAN YANG[3], CHIN-CHUN CHANG[4], (Member, IEEE), WEI-JEN WANG[1], (Member, IEEE), AND DERON LIANG[1]

[1]Department of Computer Science and Information Engineering, National Central University, Taoyuan City 320317, Taiwan
[2]College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350007, China
[3]Institute for Information Industry, Taipei 10622, Taiwan
[4]Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202301, Taiwan

Corresponding author: Deron Liang (drliang@csie.ncu.edu.tw)

This study was supported by the National Science and Technology Council of Taiwan, under grants 113-2221-E-008-109-MY2, and the Software Research Center, National Central University.

ABSTRACT In industrial production, automated inspection methods for circular knitting machines often encounter several challenges. First, the rapid movement of fabrics on these machines makes it difficult for existing fabric defect detection methods to effectively capture and process the motion. Next, due to practical constraints aimed at maintaining high yield rates, collecting sufficient abnormal fabric samples for model training is costly and limited. Furthermore, circular knitting machines typically operate under varying illumination conditions, further complicating the task of accurate fabric defect detection. Additionally, these methods usually fail to identify the cutline patterns that are integral to the design of the fabric and mistake cutlines for v-line defects. Therefore, existing fabric defect detection methods often struggle to balance real-time processing, few-shot learning, and high accuracy under various illumination conditions To address the aforementioned challenges, we adopt a few-shot learning approach and propose a novel real-time fabric defect detection method for circular knitting machines, aiming to achieve high accuracy even under varying illumination conditions. The proposed mechanism consists of two components, the LBUnet and the false alarm filter for cutlines. First, to tackle the challenges of real-time detection, limited training data, and varying illumination conditions, we develop a lightweight semantic segmentation model, LBUnet, which leverages local binary (LB) convolution to effectively handle variable lighting conditions. Next, to address the specific challenge of detecting V-line defects, we propose a false-alarm filtering method that ensures accurate defect identification by utilizing time-series data composed of consecutive segmentation maps generated by LBUnet. Extensive experiments demonstrate that the proposed method delivers both high defect detection accuracy and real-time processing performance for fast-moving fabrics on circular knitting machines under diverse lighting conditions. Specifically, using only LBUnet, our approach achieved an average Mean Intersection over Union (mIoU) of 86.24% with an average processing time of just 4 milliseconds per image. When the false-alarm filtering component was incorporated, the system achieved 100% accuracy in detecting cutlines.

INDEX TERMS Fabric defect detection, local binary convolution, circular knitting, real-time detection, few-shot learning, various illumination conditions.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Zia Ur Rahman.

I. INTRODUCTION

Defect detection is crucial for quality and efficiency in textile manufacturing because defects not only compromise quality but also lead to waste. Conventional automatic optical inspection (AOI) methods, which involve an intricate array of devices for data acquisition, including RGB sensors and illumination kits, are typically employed after production before the textiles are moved to the finished product warehouse [1]. However, these methods often require a subsequent round of manual examination for defect validation [1]. In AOI, semantic segmentation is the most commonly used technique [2]. Knitted fabric is commonly used in clothing manufacturing. To produce knit fabric, manufacturers usually use circular knitting machines. Conventional AOI methods are also suboptimal because they are only applied after the knitted fabric [3], and existing AOI methods tend to provide inconsistent segmentation performance under certain conditions in real time [2], [4], [5], [6], particularly in high-speed circular knitting machines. These methods often focus on solving specific challenges, such as real-time detection, handling complex lighting conditions, or dealing with data scarcity, but fail to address the need for accurate, real-time defect detection in fast-moving production environments.

Therefore, a real-time method for detecting defects during knitting is required. Considering the continuous nature of the knitting process and the high speed at which the machine operates, this method must be robust against data scarcity and variations in illumination conditions and textile type [7]. In particular, the method must be capable of detecting patterns that are integral to the design of the fabric (e.g., cutlines) and not mistake them for defects. In general, the real-time method often faces several key challenges during industrial production:

1) **Real-time detection**: The high-speed operation of circular knitting machines on production lines requires the accurate capture and analysis of hundreds of images per minute in real time. This imposes significant demands on the processing speed and responsiveness of the detection system.
2) **Scarcity of abnormal data**: Due to the variety and limited quantities of products, abnormal fabric samples are rare, and the annotation process is both complex and time-consuming. As a result, it is essential to reduce dependence on large volumes of labeled data to effectively support production optimization.
3) **Varying illumination conditions**: Uneven lighting in circular knitting machine environments poses a major challenge. Owing to spatial and budget constraints, it is impractical to deploy large-scale hardware such as RGB sensors and dedicated lighting systems. This results in images with inconsistent color gradients and illumination, which significantly hampers accurate fabric feature extraction.
4) **V-line defect detection**: Distinguishing between actual vertical line defects (v-lines) and intentional cutlines in knitted fabric is particularly difficult, as they appear visually similar. Preventing false alarms in such cases is crucial for reliable defect detection (as illustrated in Fig. 1).

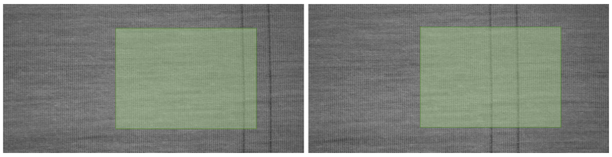

FIGURE 1. In the left panel, the photographed area (during green) includes a partially captured cutline that resembles and might be mistakenly classified as a v-line. The full cutline is shown in the right panel.

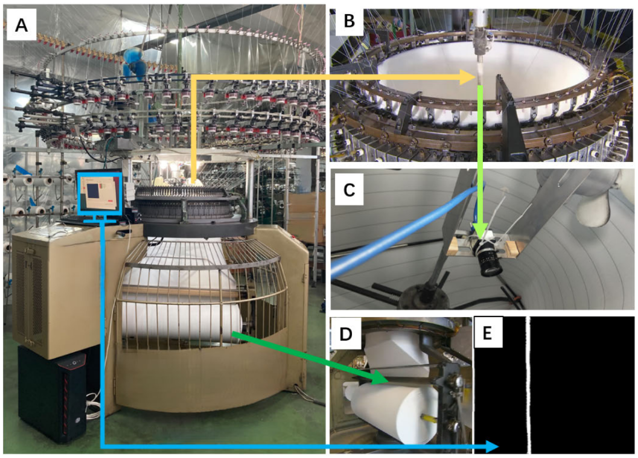

FIGURE 2. Proposed RFDC mechanism implemented in a defect detection setup: (A) Circular knitting machine and inspection mechanism; the black area highlights the monitor that displays information about the segmented defects. (B) Internal view of a knitting machine. (C) Industrial camera inside a knitting machine. (D) Finished fabric, resembling a circular pocket. (E) Segment map of the defect image displayed on the monitor.

In this paper, we propose a mechanism called Real-time Fabric defect Detection on Circular knitting machine (RFDC) to address the above challenges. In particular, we adopt a few-shot learning approach and propose a novel real-time fabric defect detection method for circular knitting machines, aiming to maintain high accuracy even under varying illumination conditions. The proposed RFDC mechanism is a comprehensive inspection solution that includes a novel lightweight deep learning model, LBUnet, which is capable of real-time semantic segmentation and a false-alarm component to realize accurate defect identification. In particular, the proposed RFDC mechanism consists of two components, the LBUnet and the false alarm filter for cutlines. First, to tackle the challenges of real-time detection, limited training samples, and varying illumination conditions, we design a lightweight semantic segmentation model, LBUnet, which incorporates local binary (LB) convolution to effectively adapt to lighting variations while maintaining high detection accuracy. Next, to address the issue of v-line defects, we propose a false-alarm filtering method that ensures accurate defect identification by leveraging time-series data composed of consecutive segmentation maps generated by LBUnet. This filtering mechanism distinguishes true defects from intentional fabric patterns, such as cutlines, thereby reducing false alarms.

To evaluate the effectiveness of the proposed RFDC, we have deployed the mechanism on a circular knitting machine, as shown on Fig. 2. This mechanism comprises the following components (Fig. 2): an edge server (Fig. 2A, bottom part) to which the captured images are sent for rapid processing and defect detection; an industrial black-and-white camera (Fig. 2C) that captures real-time images of textiles during production at a shutter speed of 30 ms; and a monitor that displays semantic segmentation maps (Fig. 2E) in real time, with problematic areas clearly marked. The proposed LBUnet was inspired by U-Net [8]. The deep learning model for knit fabric defection detection is accurate, lightweight, and robust to variable illumination conditions owing to its use of local binary convolution, a residual dense connection module, and the concatenation of multiscale features. The proposed RFDC mechanism can also distinguish between defects and cutlines in images that have been subjected to semantic segmentation. Cutlines integral to the fabric can be segmented to generate false alarms. However, the proposed mechanism continually identifies and eliminates these false alarms, which often occur at each rotation cycle of the knitting process. Experiments were conducted to verify the mechanism's effectiveness in this regard. The results revealed that the LBUnet achieved an average mean intersection over union (mIoU) of 86.08% and a false-alarm interception rate of 100%.

The primary contribution of this research is the proposed practical strategy for real-time detection of fabric defects. The RFDC mechanism offers a quantum leap in the detection of fabric defects. The main contributions of this paper are threefold:

1) The proposed LBUnet is pioneering in its application during the knitting process, providing real-time functionality while maintaining accuracy and robustness against environmental noise and various illumination conditions. It only requires 15 normal and 15 cutline samples for model training.
2) We introduce a novel dataset for knitting fabrics, comprising 16,414 images across three fabric categories under varied lighting conditions. This dataset enables research in real-time defect detection and allows for differentiating cutlines from actual defects, the dataset details are provided in the GitHub repository.[1] The proposed LBUnet model achieves an optimal 250 FPS with only 15 training samples on each category.
3) The experimental results highlight the effectiveness of the proposed mechanism, with an average mIoU of 86.08% and a 100% false-alarm interception rate for the circular knitting machine.

The remainder of this paper is organized as follows. Section II presents a literature review, Section III outlines the datasets used, Section IV details the operations of the proposed method, Section V presents the experimental results, and Section VI concludes the study.

[1] https://github.com/yokeira5/Circular_Knitting_Dataset

II. RELATED WORK

Quality inspection is an essential procedure for ensuring product quality in the fabric manufacturing industry. In the literature, there are three main topics concerning fabric defection detection: real-time defection [6], [9], [10], [11], [12], [13], [14], [15], [16], [17], few-shot learning [4], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], and various illumination and noise conditions [12], [28], [29], [30], [31], [32]. We will introduce related studies in the following subsections and then provide a summary of these related studies.

A. REAL-TIME DETECTION

Real-time detection is a critical concern in industrial inspection, with the aim of enabling real-time anomaly detection on production lines. In this context, three main kinds of efficient segmentation methods are discussed as follows:

1) Lightweight backbone. DFANet [14] and AYOLOv3-Tiny [6] employ a lightweight backbone to reduce computational costs and introduce a cross-level feature aggregation module to enhance performance. DFNet [15] uses the ‘‘Partial Order Pruning’’ algorithm to realize a lightweight backbone and an efficient decoder. RD-Unet [2] employs an attention mechanism and a simple autoencoder backbone. ICNet [16] uses a multiscale image cascade to balance speed and accuracy. Notice that, lightweight backbones often result in lower accuracy and difficulty in complex or detailed feature representation.
2) Multi-branch architecture. BiSeNetV1 [17] and BiSeNetV2 [10] introduce two-stream paths to separately capture low-level details and high-level context information. Ma et al. [13] proposed a knowledge distillation method that improves the performance of lightweight models by segmenting fabric defects to balance model efficiency and detection performance.
3) Small input resolution: RetinaNet [12] accelerates models via effective loss optimization to reduce input size and increase throughput. Although a smaller input resolution can be effective, it often overlooks detailed appearances around boundaries and small objects. In addition to choosing these strategies, the use of hardware optimization techniques [9], [11] is another common method to improve the inference speed.

B. FEW-SHOT SEGMENTATION

Few-shot segmentation (FSS) aims to create a generic model by using only a few labeled samples. Due to the limited amount of annotated data, it is essential to thoroughly extract category information from support images for FSS tasks. Mainstream FSS methods can be broadly categorized into two types based on how they extract support information: prototypical feature learning and affinity learning. Most prototypical feature learning methods [20], [21], [27] compress masked support features into one or more prototypes for feature comparison or aggregation. For

TABLE 1. Summary of the related work.

Methods	Employed Technologes	Few-shot Detection	Realtime Processing	Resistence of Illumination Noise
[20], [21], [27]	Prototype network	Yes	Not reported	No
[23], [26]	Affinity learning network	Yes	Not reported	No
[24], [25]	Large language models [24]:CLIP [25]:ViT	Yes	Not reported	No
[18], [19], [22]	Data augmentation techniques	Yes	Not reported	No
[28]	Data augmentation, Lightweight CNN-based bacnkbone	Yes	Not reported	Yes
[4]	Data augmentation, Lightweight CNN-based backbones	Yes		No
[12]	Zoom input resolution, CNN-based model	No	106.09 ms	No
[9]	GTX 1070Ti, Multi-branch CNN backbone	No	Yes (not reported)	No
[11]	SSDLiteM2-based model	No	65 FPS	No
[5], [10] [13], [17]	Multi-branch CNN architecture	No	Yes (not reported)	No
[6], [9] [14], [15]	Lightweight CNN-based backbones	No	[14]:160 FPS Others: Yes (not reported)	No
[16], [33], [34]	Lightweight CNN-based backbones, Nulti-branch CNN architecture	No	[16]: 30.3 FPS	No
[2]	Lightweight attention model	No	58 FPS	Yes
[29], [30], [32]	Image processing, CNN-based	No	No reported	Yes
[31]	Thermal techniques, CNN-Based	No	No reported	Yes

instance, SG-One [27] uses cosine similarity to compare the prototype obtained through masked average pooling with the query features for segmentation. However, relying solely on highly compressed prototypes can lead to information loss, which results in poor segmentation performance. In contrast, affinity learning methods [23] consider fine-grained pairwise relationships between the support and query features to preserve details. For example, CyCTR [26] employs a cycle-consistent attention mechanism to selectively aggregate support features. Shi et al. [23] applied attention weights between the support and query features to achieve additive aggregation of support masks. Despite achieving promising results, pixel-level correlation can suffer false matches due to intraclass variations and cluttered backgrounds. To meet the requirement of anomaly positioning without the need for real defective images, some data augmentation methods [4], [18], [19], [22] and GAN-based methods [28] synthesize pixel-level defective images. For example, LSeg [24] and GroupViT [25] integrate large language models (LLMs) with visual data to realize semantic segmentation without explicit pixel-level annotations. However, these LLM-based methods are not well suited to industrial images. In addition, recent one-class detection techniques [35], [36], [37] that leverage feature generation to simulate the missing abnormal features during training also present a promising direction worth exploring for fabric defect detection.

C. ILLUMINATION WITH NOISE

To eliminate the effects of illumination, numerous prior intrinsic image decomposition approaches have been explored. Li et al. [29] proposed an intrinsic image decomposition method that incorporates nonlocal texture constraints into traditional techniques. Shen and Yeo [30] maximized the chromatic information by assuming that neighboring pixels with similar chromaticities share the same reflectance. FabricGAN [28] enhances GAN for data augmentation in fabric defect detection by generating images under different illumination conditions with different numbers of defects to improve data diversity. However, the quality of the images generated by FabricGAN [28] is highly dependent on the training data and architecture of the GAN, which can sometimes result in less realistic effects. Zhao et al. [18] improved the algorithm based on YOLOv5 and constructed a special dataset for textile defect detection through

TABLE 2. Summary of training and test datasets that differed by texture type and illumination condition.

Dataset	Illumination Type	Texture	Training			Testing			
			Normal	Cutline	Gray Range	Normal	Cutline	Defect	Gray Range
Dataset1	Uniform illumination	Texture-1	15	15	102-117	4259	313	51	102-117
Dataset2		Texture-2	15	15	70-79	1702	81	23	70-79
Dataset3		Texture-3	15	15	36-46	4080	191	19	36-46
Dataset4	Uneven illumination	Texture-1	15	15	102-117	991	68	121	69-78
Dataset5		Texture-2	15	15	70-79	1706	102	22	152-94
Dataset6		Texture-3	15	15	36-46	1777	78	13	155-203

mirror flipping, noise injection, brightness adjustment, and other techniques. The performance of the model developed by Zhao et al. may also be limited by the diversity and representativeness of the enhanced data. RetinaNet [12] introduced a multiscale Retinex algorithm to mitigate the impact of illumination changes on fabric defect detection. This algorithm logarithmically transforms images to improve the entropy, contrast, and average gradient of image information while preserving local details. The multiscale Retinex algorithm effectively reduces this influence. Sun et al. [32] used mean filtering to eliminate noise, proposed an adaptive threshold method based on brightness to eliminate the influence of uneven illumination, and achieved foreground and background separation. Kazim Yildiz et al. [31] presented a novel thermally based fabric defect detection technique that uses a thermal camera. This approach eliminates the need for specialist operators and provides a cost-effective solution for detecting fabric defects under different illumination conditions.

D. SUMMARY

The discussed methods are summarized in Table 1. To address the issue of real-time processing, current approaches employ data augmentation [18], [19], [22] and specialized CNN-based networks, including multibranch CNN architectures [5], [9], [10], [13], [17], lightweight CNN-based backbones [2], [4], [6], [11], [14], [15], [16], [33], [34], zoom input resolution techniques [12], and GPU hardware acceleration [9]. To address the few-shot detection problem, current methods employ data augmentation [4], [18], [19], [22], [28] and specialized CNN-based networks such as prototype networks [20], [21], [27], affinity learning [23], [26], and large language models [24], [25]. In addition, to handle illumination noise, current approaches employ image processing [29], [30], [32] and thermal techniques [31]. Among the existing methods, very few are capable of handling two or more challenges. PWWU-Net [4] performs FSS and real-time detection, and RD-Unet [2] performs real-time detection and manages illumination noise. PSPNetKD [13] is a recent real-time method that can achieve 60 FPS, and FDS-Net [5] introduces innovative approaches that demonstrate outstanding performance in fabric defect detection. No existing methods address all challenges simultaneously.

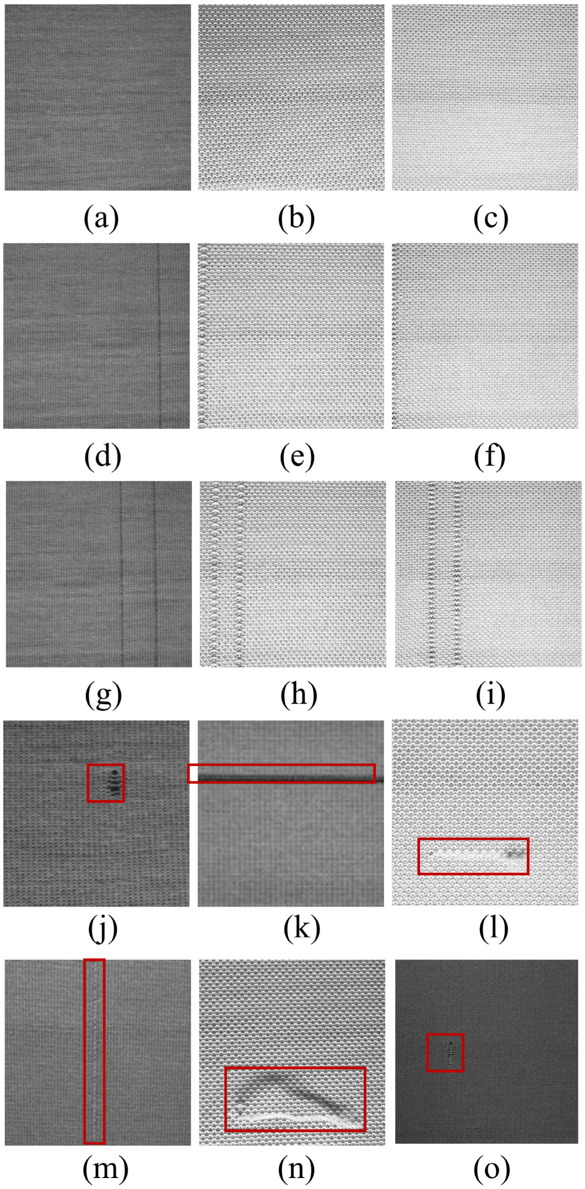

FIGURE 3. Samples of normal (defect-free, cutline, and partial-cutline) fabrics (a)-(i) and defective fabrics (j)-(r). they illustrates three types of fabric textures across various conditions. For Texture 1, images (a), (d), and (g) showcase normal (defect-free), partial-cutline, and cutline, respectively. (b), (e), and (h) correspond to Texture 2, while images (c), (f), and (i) represent Texture 3.

III. DATASETS

We trained and evaluated the proposed RFDC mechanism on six datasets that were collected using the setup shown in Fig. 2, the dataset details are provided in the GitHub repository.[2] The camera was positioned at a fixed distance

[2] https://github.com/yokeira5/Circular_Knitting_Dataset

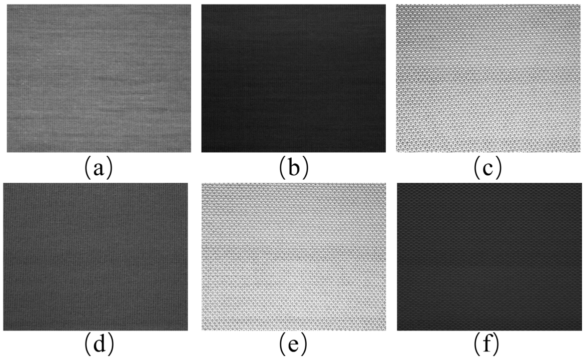

FIGURE 4. Examples of fabric images under varying illumination conditions: samples from (a) dataset-1 with gray levels ranging from 102 to 117; (b) dataset-2, 70–79; (c) dataset-3, 36–46; (d) dataset-4, 69-78; (e) dataset-5, 152–194; and (f) dataset-6, 155–203.

of approximately 15 cm from the fabric surface, and images were captured at a resolution of 640 × 800 pixels at 30 FPS. This setup ensured that the camera capture rate exceeded the knitting speed of the circular knitting machine. The industrial camera then resized the images to a resolution of 320 × 400 pixels and sent them to an edge server for processing using the proposed LBUnet for defect segmentation. Images of identified defects were then displayed on a monitor. Because each image took approximately 10–15 ms to be transmitted, each segmentation instance was completed rapidly within 15 ms. Each dataset contained images of raw white cloth with three distinct knit textures denoted as Textures 1, 2, and 3 (Fig. 3(a)-Fig. 3(c)). Images captured under uniform and uneven illumination conditions were also included to ensure that the model was robust against variations in illumination conditions.

Table 2 summarizes these datasets and their segmentation into training and testing sets for different illumination conditions. Fig. 4 presents sample images from different datasets captured under varying illumination conditions. The datasets contained 15687 images, including images of defective samples (Fig. 3). In particular, we collected six datasets: Dataset 1 (Texture 1, uniform illumination), Dataset 2 (Texture 2, uniform illumination), Dataset 3 (Texture 3, uniform illumination), Dataset 4 (Texture 1, uneven illumination), Dataset 5 (Texture 2, uniform illumination), and Dataset 6 (Texture 3, uniform illumination). Note that "uniform illumination" refers to conditions where the lighting environment remains consistent between the training and testing data, whereas "uneven illumination" indicates discrepancies in lighting conditions between training and testing, leading to inconsistencies in image appearance. All fabric was produced on the same knitting machine, and all images were captured at predefined intervals (Fig. 2C).

Each training dataset included 30 randomly selected images, including 15 images of normal fabric and 15 images of fabric with a cutline. Images of defective samples were not included because capturing these images in a rapidly changing production line is difficult [38]. In particular, the training and testing datasets did not overlap. Images in the testing dataset were collected sequentially to reflect the dynamic nature of real-world industrial processes.

IV. PROPOSED MECHANISM

The proposed RFDC mechanism is based on a lightweight deep learning detection model (LBUnet) and employs a filter-based postprocessing component with a mechanism to identify cutlines. The images were analyzed immediately after being captured. LBUnet is designed to be flexible and robust against illumination variations through transfer learning with pretrained weights. A filter-based postprocessing component exploits time-series data to refine detection results and identify cutlines to ensure accuracy. Furthermore, to prevent false alarms, the proposed RFDC mechanism distinguishes between v-lines and cutlines by subjecting the semantic segmentation results of the deep learning model to a buffer for subsequent classification. In general, v-lines and partial cutlines are difficult to distinguish from each other. This buffering approach allows for the temporal validation of potential defects, ensuring that the differentiation between a defect and (intentionally placed) cutline is both accurate and reliable. The operating flow of the RFDC mechanism, including the timing and validation criteria, is detailed in Section IV-B and is presented in Fig. 6.

A. LBUnet

The texture of a piece of fabric is defined by its repeating microstructures. A detection method must be sufficiently sensitive to local structural changes to detect subtle anomalies, such as missing or disarrayed threads. This challenge is complicated by variations in the illumination conditions under which fabric images are captured, which introduce noise and inconsistencies into the data.

LBUnet is based on U-Net [8]. A typical U-Net model has a systematic structure with downsampling and upsampling paths. For LBUnet, we halved the channel count at each level, thereby optimizing the network to focus on pertinent features for fabric textures while avoiding overfitting due to illumination variations. In the encoder, residual convolution (Fig. 5 is used to mitigate the vanishing gradient problem and enhance feature propagation without requiring several parameters. LBUnet employs local binary convolution [39] (Fig. 5, left panel) to enhance its sensitivity to different texture types and robustness against variations in illumination conditions. Local binary convolution uses binary weights and fixed nonlearnable filters to replace traditional convolutional layers, which significantly reduces the number of trainable parameters in the model. This not only enhances computational efficiency but also simplifies the network structure, effectively preventing overcomplexity in models trained on small datasets without sacrificing performance.

The LBUnet encoder (Fig. 5) operates as follows. First, a local binary convolution layer with a 1 × 1 kernel is used to expand the number of feature dimensions and capture fine textural variations. Subsequently, a series of 3 × 3 convolutions with maximum pooling are applied to

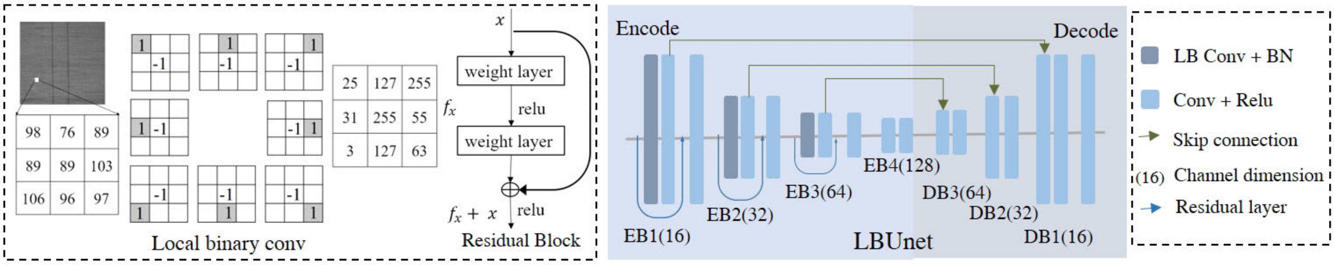

FIGURE 5. The proposed LBUnet architecture.

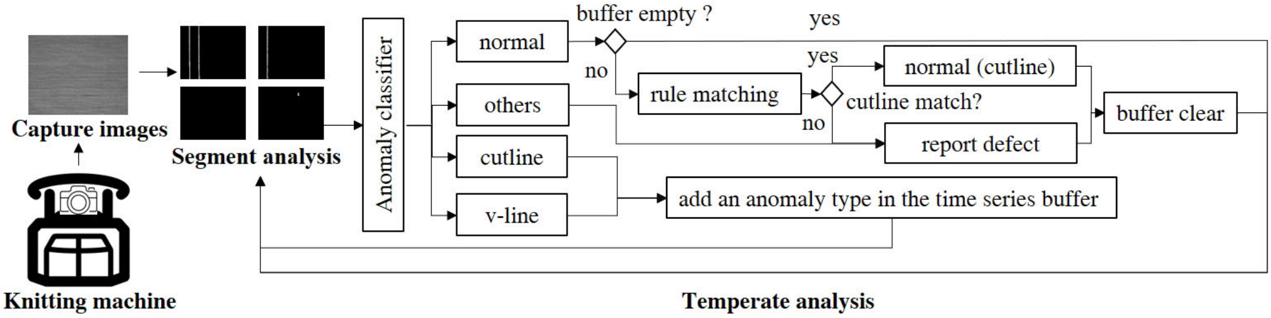

FIGURE 6. Flow of defect detection using RFDC.

downsample the feature maps to emphasize the key textural features. A residual block is added after the last convolution step, with each block comprising two weight layers with ReLU activation and an element-wise addition operation; This is repeated three times to deepen the model's feature extraction capabilities.

The LBUnet decoder (Fig. 5, right panel) operates as follows. First, deconvolution is used for upsampling to refine the feature maps, and this is complemented by repeated 3 × 3 convolutions and skip connections that merge the output of the decoder with the encoder's feature maps, thus preserving spatial information. The process culminates with processing by a convolution layer that acts on the tensor to produce an output that matches the original image size.

The aforementioned architecture reduces the LBUnet size to approximately 2.59 MB relative to that of U-Net. These computational savings give LBUnet higher throughput and enable its application in edge devices or resource-scarce environments.

B. FILTER-BASED POSTPROCESSING COMPONENT

The proposed RFDC mechanism operates as follows (Fig. 6). First, LBUnet performs semantic analysis to identify potential anomalies. For each input image, the mechanism creates a semantic segmentation map in which all defects and cutlines are marked. As mentioned, cutlines do not represent defects but closely resemble v-line defects (when captured partially). Thus, the mechanism must determine whether a marked segmentation map contains only a cutline. A filter-based component is then used to distinguish between cutlines and v-line defects based on time-series data that contain a sequence of incoming segmentation maps. The proposed mechanism categorizes fabric features in a segmentation map into one of the following four classes:

- Cutline class: A segmentation map with only a full-view cutline is categorized as a "cutline" class by the filter-based postprocessing component.
- V-line class: A segmentation map that contains only a v-line defect or a partial view of a cutline is categorized as a "v-line" class. This categorization is tentative; the filter does not report this categorization immediately. Instead, the categorization result is temporarily stored in a buffer for verification through additional data.
- Normal class: A segmentation map with no marks is categorized as a "normal" class.
- "Others" class: A segmentation map with any marked defect (excluding v-line defects) is categorized as an "others" class, which is considered defective.

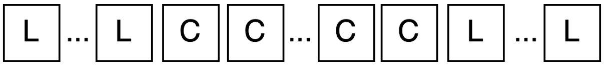

FIGURE 7. Captured cutline images in chronological order, where "L" refers to a v-line image and "C" a cutline image.

The identification of cutlines is governed by rules based on predictable cutline patterns in the knitting process; specifically, cutlines are always preceded and succeeded by a v-line. Thus, a time series of cutline images always has a continuous sequence of v-line segmentation maps, followed by a set of several cutline segmentation maps and then one or more v-line segmentation maps. Owing to the varying rotational speed of the knitting machine during each rotation cycle, the number of cutlines is greater than the number of v-lines (Fig. 7). This time-series pattern sets

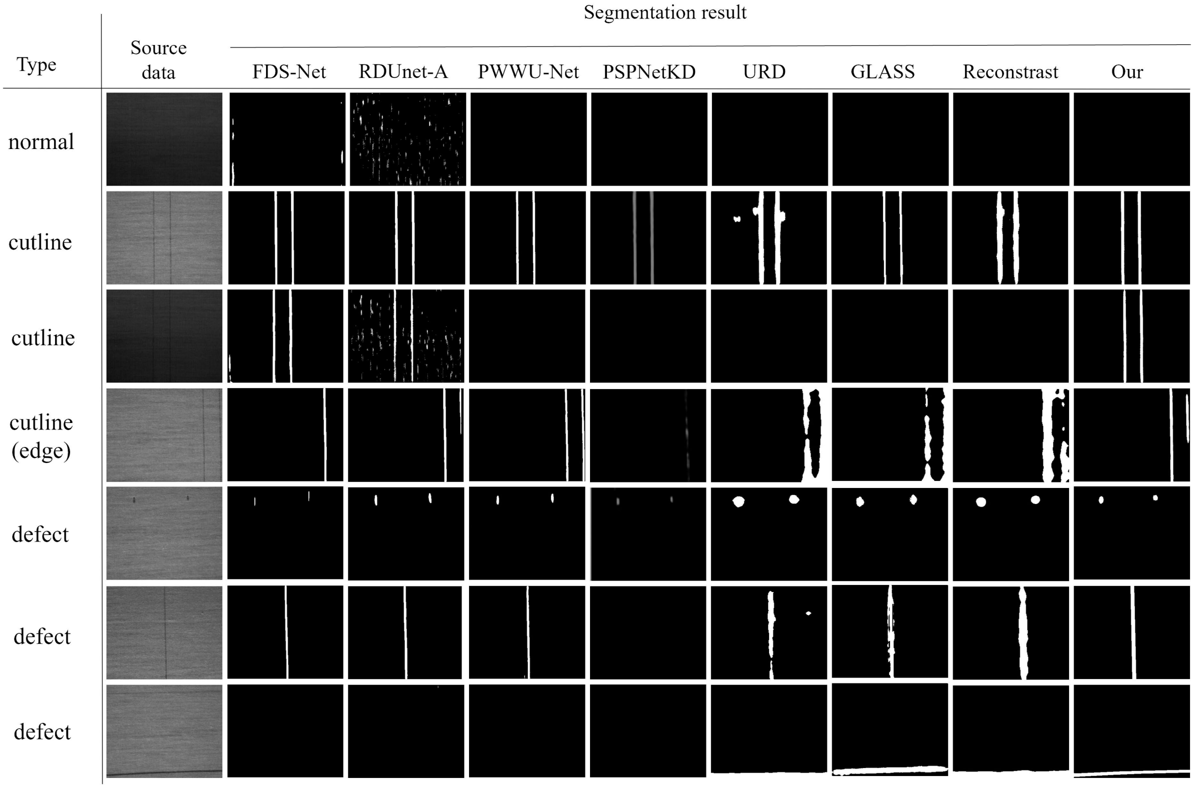

FIGURE 8. Examples of images captured under six illumination levels of fabrics of different textures and with different defects. Original fabric images (first column) and semantic segmentation results (second to fifth columns). The semantic segmentation results highlight the defects detected in each fabric sample.

(intentionally designed) cutlines apart from defective v-lines. When the RFDC identifies either a cutline or a v-line, the corresponding segmentation maps are stored in a buffer. The content of the buffer is retrieved by the filter-based component when the component detects a normal class. Subsequently, the retrieved buffer content is assessed to determine whether the aforementioned cutline pattern is present (Fig. 7). If the cutline is normal, the component clears the buffer and continues detection. However, if the component determines that a v-line defect or some other defect is present, the mechanism issues an alarm, and the monitor displays a segmentation map for the defective image (Fig. 2(E)).

To classify a segmentation map into the normal, cutline, v-line, and "others" classes, the RFDC mechanism projects the 2D detected segmentation result onto the 1D vector by summing the vertical pixels of the detected segmentation result. In a segmentation map, 1 and 0 represent marked and normal pixels, respectively. Specifically, the projection from a 2D segment map A of size $N \times M$ onto a 1D vector V of size M is as follows:

$$v_j = \sum_{i=0}^{N-1} a_{ij} \tag{1}$$

where a_{ij} is the pixel (i,j) of A and v_j is the jth element of V. Subsequently, the sum of all pixels in the segmentation map is calculated as follows:

$$S = \sum_{j=0}^{M-1} v_j \tag{2}$$

where S is the sum of all pixels in the segmentation map. S is then roughly classified as normal or non-normal (cutline, v-line, and others) as follows:

$$\hat{y} = \begin{cases} 0 \text{ (Normal)}, & \text{if} \quad S = 0; \\ 1 \text{ (Non-normal)}, & \text{otherwise} \end{cases} \tag{3}$$

where $\hat{y}$ is the initial classification for the segmentation map and $\hat{y} = 0$ and $\hat{y} = 1$ correspond to the normal and non-normal (cutline, v-line, and others) classes, respectively. If $\hat{y} = 1$, further classification is required (Fig. 8). To distinguish between cutlines, v-lines, and other defects, the RFDC calculates the mean and variance of the 1D vector V to estimate the dispersion of the segmentation map, as follows:

$$\mu = \sum_{j=0}^{M-1} \frac{v_j}{M} \tag{4}$$

$$\sigma = \sqrt{\sum_{j=0}^{M-1} \frac{(v_j - \mu)^2}{M}} \tag{5}$$

where μ and σ are the mean and variance, respectively, of the 1D vector V. As mentioned, the RFDC distinguishes cutlines from other defects based on the time-series cutline pattern; it does so using the variance of the projected 1D vector of the segmentation map as follows:

$$\hat{y} = \begin{cases} 0 \text{ (Cutline)}, & \text{if} \quad F_c \leq \sigma < F_d; \\ 1 \text{ (V-line)}, & \text{if} \quad \sigma < F_c; \\ 1 \text{ (Defect)}, & \text{otherwise}. \end{cases} \tag{6}$$

where $\hat{y} = 0$ indicates that the fabric is normal with a cutline, and $\hat{y} = 1$ indicates that the fabric is non-normal. $F_c \leq \sigma < F_d$ indicates the presence of a cutline, and $\sigma < F_c$ indicates the presence of a v-line. Eq. 6 uses two predefined values, F_c and F_d, which calculated from the training cutline datasets, F_c is minimum variance obtained from the segment maps in thr training datasets, while F_d the maximum variance. During inference, we assess whether the non-normal σ falls within this cutline variance range to determine if it corresponds to a cutline.

After the RFDC has classified all instances, it places cutline and v-line results into a buffer and retrieves them only when a normal instance is detected. Subsequently, the proposed mechanism analyzes the buffer data to determine whether they show a sequence of continuous segmentation maps taken from the cutline images (full or partial view). Specifically, the mechanism applies a rule that states that a buffer image is normal if and only if it (1) is flanked by v-lines at both sides and (2) has more v-lines than cutlines.

V. EXPERIMENTAL RESULTS

We used the six datasets described in Section III to evaluate the performance of the proposed LBUnet relative to that of four real-time and state-of-the-art fabric AOI methods: RD-Unet [2], PWWU-Net [4], U-net [8],PSPNetKD [13], FDS-Net [5], GLASS [40], URD [41], and GeneralAD [42]. The proposed mechanism was noted to be the only one that could filter false alarms during knitting. The experiments were conducted on a system with an Nvidia GeForce RTX 3090 graphics card. The training optimizer was Adam with a learning rate of 0.001. The images were processed at a batch size of 8. We conducted two experiments under different illumination conditions: one under uniform illumination and the other under uneven illumination.

A. DATASET

We add a few-shot segmentation dataset as experiment dataset, FSSD-12 [43], to address the severely insufficient pixel-wise labeled defect samples in existing works. there are twelve classes in FSSD-12, including abrasion-mask, iron-sheet ash, liquid, oxide-scale, oil-spot, water-spot, patch, punching, red-iron sheet, roll-printing, scratch, and inclusion, samples as shown in Fig. 9.

B. EXPERIMENTAL FOR UNIFORM ILLUMINATION

This experiment used three datasets: Dataset 1 for Texture 1, Dataset 2 for Texture 2, and Dataset 3 for Texture 3. Because all images in the three datasets were captured under uniform illumination, the AOI method could easily identify defects.

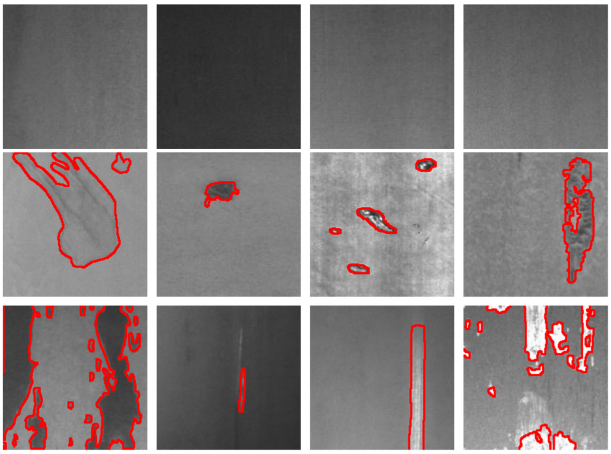

FIGURE 9. Samples of defect images from FSSD-12 [43] (defect area marked in red).

C. EXPERIMENTAL FOR UNEVEN ILLUMINATION

This experiment used Dataset 4 for Texture 1, Dataset 5 for Texture 2, and Dataset 6 for Texture 3. For each dataset, the illumination conditions of the testing group differed from those of the training group. The reason for using varying illumination conditions was that we intended to simulate different illumination conditions during model training and inference. This should be common in industrial production environments.

D. EVALUATION METRICS

All methods were evaluated on the same datasets for a fair comparison. The semantic segmentation performance was evaluated using the mIoU as follows [43]:

$$mIoU = \frac{1}{k} \sum_{i=0}^{k-1} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{7}$$

where TP_i is the number of true-positive pixels in class i, FP_i is the number of false-positive pixels, FN_i is the number of false-negative pixels, and k is the number of classes (i.e., normal and defective). We evaluated the speed of the methods in terms of FPS, number of parameters, and accuracy.

E. COMPARISON RESULTS

1) EVALUATION OF DEFECT SEGMENTATION PERFORMANCE

First, we compare the performance of various methods on the proposed dataset. To ensure a fair comparison, we adopt official or widely recognized implementations of each method and evaluate them under consistent experimental settings—such as using the same training and testing splits from our custom dataset, the same number of training epochs, and identical hyperparameters. Furthermore, we assess each method's performance under different experimental

TABLE 3. Fabric segmentation results obtained using different AOI deep learning methods on training and testing datasets at the same illumination level.

Uniform Illumination	Method	Normal	Cutline	Defect	Average
		mIoU ↑			
Dataset1	PWWU-Net [4]	100	57.4	60.3	72.6
	PSPNetKD [13]	100	55.8	45.3	67.1
	U-net [8]	100	55.8	48.2	68.0
	FDS-Net [5]	100	84.9	71.4	85.4
	RD-Unet [2]	100	85.1	61.6	82.2
	GLASS [40]	100	74.8	54.8	76.5
	URD [41]	100	73.3	47.2	73.5
	GeneralAD [42]	100	63.8	61.4	75.1
	Ours	100	86.2	71.4	85.8
Dataset2	PWWU-Net [4]	100	71.3	75.8	82.3
	PSPNetKD [13]	100	62.3	40.2	67.5
	U-net [8]	100	55.5	54.4	69.9
	FDS-Net [5]	100	81.3	71.6	84.3
	GLASS [40]	100	60.8	59.2	73.3
	URD [41]	100	53.6	56.4	70.0
	GeneralAD [42]	100	62.2	65.0	75.7
	RD-Unet [2]	100	85.6	59.0	81.5
	Ours	100	87.1	71.4	86.2
Dataset3	PWWU-Net [4]	100	52.4	54.0	68.8
	PSPNetKD [13]	100	48.2	47.9	65.3
	U-net [8]	100	55.3	52.1	69.1
	FDS-Net [5]	100	84.1	60.7	81.6
	RD-Unet [2]	100	86.8	62.6	83.1
	GLASS [40]	100	58.1	58.3	72.1
	URD [41]	100	58.4	55.2	71.2
	GeneralAD [42]	100	63.0	62.9	75.3
	Ours	100	87.0	73.3	86.7
Overall	PWWU-Net [4]	100	60.4	63.4	74.6
	PSPNetKD [13]	100	55.5	44.5	66.6
	U-net [8]	100	55.5	51.6	69.0
	FDS-Net [5]	100	83.4	67.9	83.8
	RD-Unet [2]	100	77.5	61.2	79.6
	GLASS [40]	100	62.2	56.5	72.9
	URD [41]	100	64.6	55.8	73.5
	GeneralAD [42]	100	70.8	61.1	77.3
	Ours	100	86.7	72.0	86.2

TABLE 4. Fabric segmentation results under uneven illumination.

Uneven Illumination	Method	Normal	Cutline	Defect	Average
		mIoU ↑			
Dataset4	PWWU-Net [4]	49.6	60.2	68.0	59.2
	PSPNetKD [13]	48.4	59.0	56.2	54.6
	U-net [8]	100	48.6	44.5	64.4
	FDS-Net [5]	60.8	49.7	49.7	53.4
	RD-Unet [2]	49.1	51.0	54.4	51.5
	GLASS [40]	58.5	55.3	53.6	55.8
	URD [41]	44.4	40.3	40.2	41.6
	GeneralAD [42]	50.3	51.8	42.3	48.1
	Ours	100	73.5	71.9	81.8
Dataset5	PWWU-Net [4]	48.4	59.0	58.2	55.2
	PSPNetKD [13]	46.0	45.6	47.2	46.3
	U-net [8]	100	48.8	49.0	65.9
	FDS-Net [5]	49.1	46.6	47.0	47.6
	RD-Unet [2]	48.8	47.7	46.2	47.6
	GLASS [40]	55.8	59.2	58.4	57.8
	URD [41]	41.6	54.5	45.9	47.3
	GeneralAD [42]	48.1	51.1	52.1	50.4
	Ours	100	69.0	65.9	78.3
Dataset6	PWWU-Net [4]	53.9	47.2	47.2	49.4
	PSPNetKD [13]	49.3	46.3	47.9	47.8
	U-net [8]	100	53.1	45.4	66.2
	FDS-Net [5]	49.4	46.7	46.2	47.4
	RD-Unet [2]	48.4	59.0	56.0	54.4
	GLASS [40]	57.8	62.9	66.4	62.4
	URD [41]	48.2	52.3	56.1	52.2
	GeneralAD [42]	54.8	60.8	59.2	58.3
	Ours	100	76.6	78.5	85.0
Overall	PWWU-Net [4]	50.6	55.5	57.8	54.6
	PSPNetKD [13]	47.9	50.3	50.4	49.6
	U-net [8]	100	50.2	46.3	65.5
	FDS-Net [5]	53.1	47.7	47.6	49.5
	RD-Unet [2]	48.7	52.5	52.2	51.1
	GLASS [40]	57.4	59.1	59.5	58.7
	URD [41]	44.7	49.0	47.4	47.1
	GeneralAD [42]	51.1	54.6	51.2	52.3
	Ours	100	73.0	72.1	81.7

scenarios, including both uniform and uneven illumination conditions.

Under uniform illumination conditions, in terms of the average mIoU, LBUnet outperformed RD-Unet by 3.92%, FDS-Net by 5.06%, U-net by 11.32%, PSPNetKD by 19.61%, and PWWU-Net by 11.25% (Table 3). PWWU-Net and RD-Unet performed well in detecting cutlines (Fig. 8). FDS-Net was generally capable of detecting defects but could not accurately identify hole defects (“Defect” row in Fig. 8). PSPNetKD [13] employs knowledge distillation to address the real-time processing challenges, resulting in improved speed. However, this approach compromises the precision of identifying and locating anomalies or defects.

Under uneven illumination conditions (Table 4), FDS-Net [5] and RD-Unet [2] misclassified the edges and low-contrast areas of the images as defects (Fig. 8, second and fourth rows). PWWU-Net [4] and PSPNetKD [13] tended to incorrectly identify all unevenly illuminated images as normal. In contrast, the proposed model provided accurate classifications with few false positives and precise delineations of defective regions.

Next, we further evaluate the generalization capability of the methods by conducting experiments on publicly available open datasets, as shown in Table 5. Although recent approaches employ various anomaly synthesis techniques—such as GLASS [40], which combines feature- and image-level synthesis; GeneralAD [42], which generates patch-wise pseudo anomalies using Vision Transformers; and URD [41], which guides reconstruction via synthesized anomaly regions—these methods still yield unsatisfactory results on the FSSD-12 dataset [43], as indicated by their lower mIoU scores in segmentation tasks. In contrast, the proposed method achieves significantly better performance, demonstrating strong generalization by producing more accurate and compact segmentation results, as evidenced by its higher mIoU scores on the FSSD-12 dataset [43].

F. ABLATION STUDY

1) IMPACT OF DIFFERENT MODULE COMBINATIONS

In Table 6, we analyze the impact of different module combinations on performance under the protocol defined for Dataset 1. First, comparing Unet + Naive prediction with LBUnet + Naive prediction, we observe that LBUnet—an enhanced version of Unet incorporating local binary (LB) convolution—demonstrates superior robustness to lighting variations, resulting in improved segmentation performance under varying illumination conditions. Next, comparing LBUnet + Naive prediction with LBUnet + Filter-based detection, we find that the proposed filter-based detection

TABLE 5. Performance comparison on the FSSD-12 dataset [43] using the mIoU metric.

Category	Steel_Am	Steel_Ia	Steel_Ld	Steel_Op	Steel_Os	Steel_Pa	Steel_Pk	Steel_Ri	Steel_Rp	Steel_Sc	Steel_Se	Steel_Ws	Average
	mIoU ↑												
GLASS [40]	47.2	61.4	53.0	57.7	50.0	59.3	53.6	62.2	57.8	49.8	51.7	44.6	54.0
URD [41]	56.4	65.0	62.6	60.9	62.6	51.3	58.4	63.0	56.9	49.4	56.5	65.1	59.0
GeneralAD [42]	57.2	62.9	66.4	55.5	62.4	62.3	53.7	62.6	61.9	54.6	59.5	65.6	60.4
Ours	59.9	63.6	65.7	63.1	64.3	69.7	55.8	63.5	60.5	57.7	65.7	64.8	**62.9**

TABLE 6. Ablation study on different module combinations on the protocol Dataset 1.

Architecture		Detection method		Normal	Cutline	defect
Unet	LBUnet	Niave prediton	Filter-based detection	Accuracy ↑		
✓		✓		100	55.84	100
	✓	✓		100	73.84	100
	✓		✓	100	100	100

TABLE 7. Ablation study on different network architectures using the protocol Dataset 1.

Network components							Normal	Cutline	Defect	FPS	Params
EB1	EB2	EB3	EB4	DB3	DB2	DB1	mIoU ↑				
✓						✓	21.52	32.21	35.81	445.63	100.3 k
✓	✓				✓	✓	66.32	38.59	45.21	343.96	444.19 k
✓	✓	✓		✓	✓	✓	100	75.46	48.65	269.94	1.82 M
✓	✓	✓	✓	✓	✓	✓	100	86.15	71.37	250.12	2.59M

method enables more accurate defect identification by leveraging time-series information from consecutive segmentation maps. This false-alarm filtering mechanism effectively differentiates actual defects from intentional fabric patterns, such as cutlines, thereby reducing false alarms.

2) IMPACT OF DIFFERENT NETWORK LAYERS

In Table 7, we examine how different network architecture configurations affect performance and inference speed under the protocol of Dataset 1. Here, EB and DB refer to the encoder blocks and decoder blocks of the proposed LBUnet, as illustrated in Fig. 5. Initially, when using only EB1 and DB1, the model achieves extremely high FPS but fails to accurately detect normal samples, cutlines, and defects, leading to poor overall performance. As additional encoder and decoder blocks are included, we observe a consistent improvement in detection performance, accompanied by a gradual decrease in FPS. Given that 250 FPS is more than sufficient for real-time industrial applications, we adopt the configuration with EB1 + EB2 + EB3 + EB4 + DB3 + DB2 + DB1 as the default model. Furthermore, this speed-accuracy trade-off can be flexibly adjusted to meet the specific requirements of different production environments. In particular, to meet the minimum requirements of production environments, the deployed fabric detection model must achieve a speed of at least 30 FAS per second in real-world scenarios, completing image acquisition, transmission, analysis, and display within 30 milliseconds.

3) IMPACT OF DIFFERENT NUMBER OF TRAINING SAMPLES

In Table 8, we investigate the impact of using varying numbers of normal and cutline samples on performance under the protocol of Dataset 1. Initially, with only a few training samples (e.g., 5 normal and 5 cutline samples), the proposed method performs poorly in detecting normal, cutline, and defect samples due to the extremely limited training data. As the number of training samples increases, performance improves gradually. Moreover, when the number of normal and cutline samples exceeds 15 each, we observe that performance begins to plateau. Considering practical application scenarios, where a balance between the amount of training data and detection performance is essential, we adopt 15 normal samples and 15 cutline samples for training based on the findings from this ablation study.

TABLE 8. Ablation study on different normal and cutline samples under the protocol Dataset 1.

Normal samples	Cutline samples	Normal	Cutline	Defect
		mIoU ↑		
5	5	64.52	55.64	51.23
10	10	100.00	70.23	65.23
15	15	100.00	86.15	71.37
20	20	100.00	88.23	75.62
25	25	100.00	84.51	85.21
30	30	100.00	86.21	79.25

4) COMPUTATIONAL SPEED

LBUnet also exhibited the fastest computational speed among the compared methods (Table 9). It had 247% fewer parameters than the best of the other four methods, which is PSPNetKD [13], and a 4.3-fold higher average FPS than did the best of the other methods. Therefore, LBUnet provided superior performance at significantly lower computational costs than the compared methods.

5) SIMULATION OF REAL-TIME DEFECT DETECTION DURING KNITTING

Fig. 10 shows time-series fabric data examples. The proposed RFDC mechanism is the first method capable of defect detection during knitting. Thus, it could not be evaluated

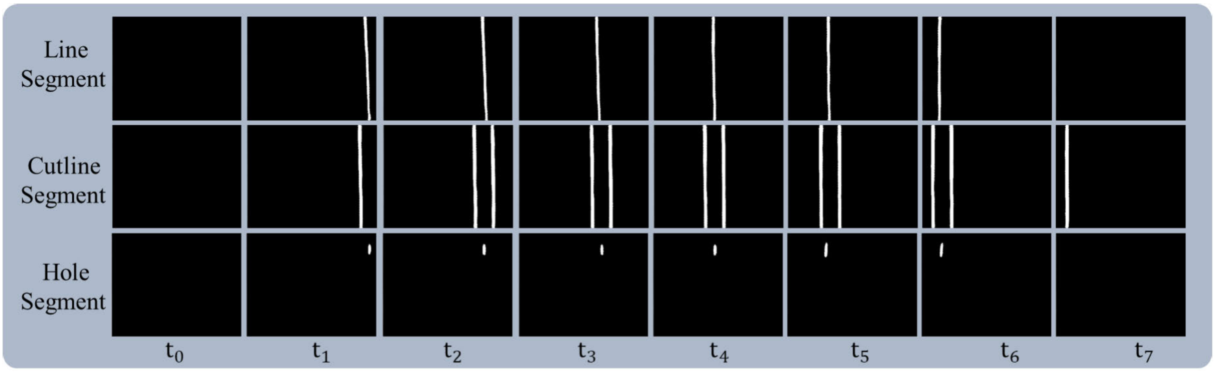

FIGURE 10. Time-series fabric data examples for defect detection in the dataset. Columns t_0 to t_7 represent different time points. Rows show images with cutlines, line defects, and hole defects. At some time points, such as t_0, cutlines and line defects have the same features, which could lead to false alarms during classification.

TABLE 9. Computational complexity (measured in terms of FPS and number of parameters) of proposed LBUnet and the three compared methods.

Method	Params size (byte) ↓	Processing speed (FPS) ↑
PWWU-Net [4]	68M	42
RD-Unet [2]	147M	12
U-net [8]	158M	10
FDS-Net [5]	117M	24
PSPNetKD [13]	54M	64
GLASS [40]	224M	10
URD [41]	95M	43
GeneralAD [42]	145M	36
Proposed	2.59M	250

TABLE 10. Fabric segmentation under uneven illumination.

Dataset	Filter-based detection	Normal	Cutline	Defect	Average
		Accuracy↑			
Dataset4	✗	100%	73.84%	100%	91.28%
	✓	100%	**100%**	100%	100%
Dataset2	✗	100%	76.56%	100%	92.19%
	✓	100%	**100%**	100%	100%
Dataset3	✗	100%	73.65%	100%	91.22%
	✓	100%	**100%**	100%	100%
Dataset4	✗	100%	91.04%	100%	97.01%
	✓	100%	**100%**	100%	100%
Dataset5	✗	100%	77.97%	100%	92.66%
	✓	100%	**100%**	100%	100%
Dataset6	✗	100%	95.00%	100%	98.33%
	✓	100%	**100%**	100%	100%
Overall	✗	100%	81.34%	100%	93.78%
	✓	100%	**100%**	100%	**100%**

against the other three methods. Only the filtering efficiency of the proposed mechanism can be evaluated. To simulate the image capture process during knitting, we fed each image from the testing group of a dataset into the RFDC mechanism in chronological order. To evaluate the ability of the proposed RFDC mechanism to prevent false alarms, two sets of experiments were conducted using the six datasets: one experiment involved applying filter-based defect detection, and the other did not involve applying such detection. The proposed filter-based postprocessing component significantly enhanced the accuracy of distinguishing between actual defects and cutlines (Table 10).

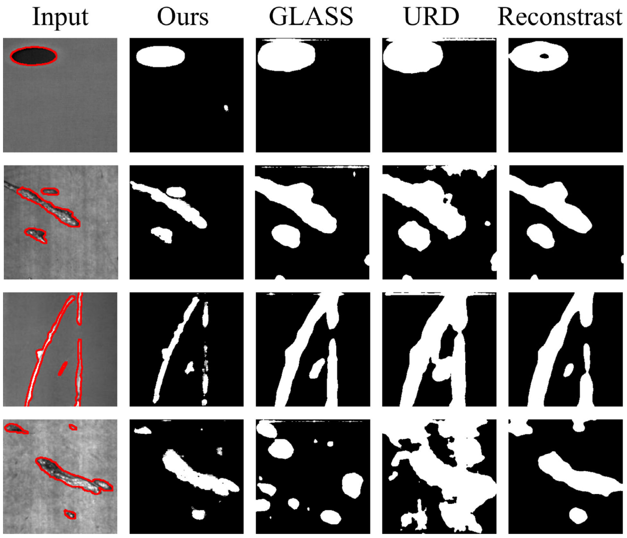

FIGURE 11. Examples of fabric defects (Defect area marked in red) alongside their corresponding segmentation results (Defect area marked in white), the first row shows simple defect samples and the second to fourth rows present complex defects.

6) VISUALIZATION OF FABRIC DEFECT DETECTION

In Fig. 11, we compare the segmentation results on fabric defect images using several recent methods. Starting with the first row, which contains samples of simple defects, we observe that other methods tend to incorrectly classify normal regions as defective, while the proposed method accurately and compactly highlights only the defective regions. In the second to fourth rows, which display samples of more complex defects, the proposed method consistently demonstrates precise and reliable detection performance. It accurately and compactly identifies defective regions, maintaining the same high-quality segmentation results as seen with the simple defects.

VI. CONCLUSION

We have introduced a mechanism for Real-time Fabric defect detection on Circular knitting machine (RFDC). The proposed mechanism is based on a deep learning segmentation model, LBUnet; it also features a false-alarm filter that prevents cutlines from being mistakenly classified as defects when captured partially or fully by a camera. First, to enable real-time deployment for industrial fabric detection, we propose a lightweight LBUnet with only 2.59M parameters. Next, we replace standard convolution layers with local binary convolution within LBUnet to enhance the model's robustness to illumination variations during the inference stage. Hence, LBUnet not only achieves high-speed response but also demonstrates strong robustness to illumination variations. In addition, the proposed mechanism handles time-series data and different types of fabric textures well. The experimental results indicated that the proposed mechanism exhibited strong performance when applied to data for three types of fabric and six illumination conditions. Therefore, the proposed mechanism is suitable for use in complex industrial environments. The speed of the proposed model mechanism was determined to be 250 FPS, which far exceeds the industry minimum of 25 FPS for AOI; it also exceeded those of competing AOI methods [2], [4], [8], [13], [40], [41], [42] by at least 430%. The RFDC mechanism achieved an average mIoU of 86.25% under uniform illumination and 81.70% under uneven illumination, outperforming models such as PSPNetKD, PWWU-Net, U-Net, FDS-Net, GLASS, URD, GeneralAD, and RD-Unet (whose mIoU ranged from 65.49% to 82.66%). Moreover, the RFDC demonstrated superior robustness to variations in lighting conditions and successfully filtered out all false alarms. Future research can explore applying the RFDC to other domains or further enhancing its architecture for improved performance.

REFERENCES

[1] Y. Xi, Y. Zhang, S. Ding, and S. Wan, "Visual question answering model based on visual relationship detection," *Signal Process., Image Commun.*, vol. 80, Feb. 2020, Art. no. 115648.

[2] H. Chen, D. Chen, and H. Dai, "RDUnet—A: A deep neural network method with attention for fabric defect segmentation based on autoencoder," in *Proc. IEEE Int. Conf. Artif. Intell. Ind. Design (AIID)*, May 2021, pp. 134–139.

[3] G. M. Nasira and P. Banumathi, "An intelligent system for automatic fabric inspection," *Asian J. Inf. Technol.*, vol. 13, no. 6, pp. 308–312, 2014.

[4] Z. Wang and J. Jing, "Pixel-wise fabric defect detection by CNNs without labeled training data," *IEEE Access*, vol. 8, pp. 161317–161325, 2020.

[5] Y. Huang, J. Jing, and Z. Wang, "Fabric defect segmentation method based on deep learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.

[6] J. Yao and J. Li, "AYOLOv3-tiny: An improved convolutional neural network architecture for real-time defect detection of PAD light guide plates," *Comput. Ind.*, vol. 136, Apr. 2022, Art. no. 103588.

[7] M. I. Bin Roslan, Z. Ibrahim, and Z. A. Aziz, "Real-time plastic surface defect detection using deep learning," in *Proc. IEEE 12th Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, May 2022, pp. 111–116.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.-MICCAI*, Munich, Germany, 2015, pp. 234–241.

[9] Y. Dai, J. Wang, J. Li, and J. Li, "PDBNet: Parallel dual branch network for real-time semantic segmentation," *Int. J. Control, Autom. Syst.*, vol. 20, no. 8, pp. 2702–2711, Aug. 2022.

[10] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, "BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3051–3068, Nov. 2021.

[11] H. Fan, S. Liu, M. Feriane, H.-C. Ng, Z. Que, S. Liu, X. Niu, and W. Luk, "A real-time object detection accelerator with compressed SSDLite on FPGA," in *Proc. Int. Conf. Field-Program. Technol. (FPT)*, Dec. 2018, pp. 14–21.

[12] A. Demidovskij, A. Tugaryov, M. Fatekhov, E. Aidova, E. Stepyreva, M. Shevtsov, and Y. Gorbachev, "Accelerating object detection models inference within deep learning workbench," in *Proc. Int. Conf. Eng. Emerg. Technol. (ICEET)*, Oct. 2021, pp. 1–6.

[13] C. Ma, L. Qi, Y. Wang, W. Wang, and Y. Sun, "Improving lightweight model for fabric defect segmentation via contrastive knowledge distillation," *Pro. SPIE*, vol. 12791, pp. 404–409, Oct. 2023.

[14] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9522–9531.

[15] X. Li, Y. Zhou, Z. Pan, and J. Feng, "Partial order pruning: For best speed/accuracy trade-off in neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9145–9153.

[16] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 418–434.

[17] C. Yu, J. Wang, C. Peng, C. Gao, and G. Yu, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 325–341.

[18] Z. Zhao, X. Ma, X. Yang, and F. Wang, "Research on fabric defect detection algorithm based on improved YOLOv5," in *Proc. IEEE 7th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2024, pp. 387–391.

[19] V. Sampath, I. Maurtua, J. J. Aguilar Martín, A. Iriondo, I. Lluvia, and G. Aizpurua, "Intraclass image augmentation for defect detection using generative adversarial neural networks," *Sensors*, vol. 23, no. 4, p. 1861, Feb. 2023.

[20] G. Li, V. Jampani, L. Sevilla-Lara, D. Sun, J. Kim, and J. Kim, "Adaptive prototype learning and allocation for few-shot segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8334–8343.

[21] Y. Liu, X. Zhang, S. Zhang, and X. He, "Part-aware prototype network for few-shot semantic segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.-ECCV*, Aug. 2020, pp. 142–158.

[22] S. Niu, Y. Peng, B. Li, and X. Wang, "A transformed-feature-space data augmentation method for defect segmentation," *Comput. Ind.*, vol. 147, May 2023, Art. no. 103860.

[23] X. Shi, D. Wei, Y. Zhang, D. Lu, M. Ning, J. Chen, K. Ma, and Y. Zheng, "Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 151–168.

[24] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," 2022, *arXiv:2201.03546*.

[25] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, and X. Wang, "GroupViT: Semantic segmentation emerges from text supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18113–18123.

[26] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-shot segmentation via cycle-consistent transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 21984–21996.

[27] X. Zhang, Y. Wei, Y. Yang, and T. S. Huang, "SG-one: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3855–3865, Sep. 2020.

[28] Y. Xu, C. Zhi, S. Wang, J. Chen, R. Sun, Z. Dong, and L. Yu, "FabricGAN: An enhanced generative adversarial network for data augmentation and improved fabric defect detection," *Textile Res. J.*, vol. 94, nos. 15–16, pp. 1771–1785, Aug. 2024.

[29] L. Shen, P. Tan, and S. Lin, "Intrinsic image decomposition with non-local texture cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[30] L. Shen and C. Yeo, "Intrinsic images decomposition using a local and global sparse representation of reflectance," in *Proc. CVPR*, Jun. 2011, pp. 697–704.

[31] K. Yildiz, A. Buldu, M. Demetgul, and Z. Yildiz, "A novel thermal-based fabric defect detection technique," *J. Textile Inst.*, vol. 106, no. 3, pp. 275–283, Mar. 2015.

[32] G. Sun, H. Li, X. Dai, D. Zhao, and W. Feng, "Method of mesh fabric defect inspection based on machine vision," *J. Engineered Fibers Fabrics*, vol. 8, no. 2, Jun. 2013, Art. no. 155892501300800213.

[33] W. Song, D. Lang, J. Zhang, M. Zheng, and X. Li, "Textile defect detection algorithm based on the improved YOLOv8," *IEEE Access*, vol. 13, pp. 11217–11231, 2025.

[34] H. Lv, H. Zhang, M. Wang, J. Xu, X. Li, and C. Liu, "Hyperspectral imaging based nonwoven fabric defect detection method using LL-YOLOv5," *IEEE Access*, vol. 12, pp. 41988–41998, 2024.

[35] P. Huang, J.-X. Chong, J. Chiang, T. Chen, T.-L. Liu, and C.-T. Hsu, "SLIP: Spoof-aware one-class face anti-spoofing with language image pretraining," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, 2025, pp. 3697–3706.

[36] P.-K. Huang, C.-H. Chiang, T.-H. Chen, J.-X. Chong, T.-L. Liu, and C.-T. Hsu, "One-class face anti-spoofing via spoof cue map-guided feature learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 277–286.

[37] P.-K. Huang, J.-X. Chong, M.-T. Hsu, F.-Y. Hsu, Y.-T. Lin, K.-H. Chien, H.-C. Shao, and C.-T. Hsu, "Unsupervised feature disentanglement and augmentation network for one-class face anti-spoofing," 2025, *arXiv:2503.22929*.

[38] C.-C. Chen, C. H. Wei, and C.-S. Lin, "Fast detection of fabric defects based on neural networks," in *Proc. 6th Int. Symp. Comput., Consum. Control (IS3C)*, Jun. 2023, pp. 322–325.

[39] F. Juefei-Xu, V. N. Boddeti, and M. Savvides, "Local binary convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4284–4293.

[40] Q. Chen, H. Luo, C. Lv, and Z. Zhang, "A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 37–54.

[41] X. Liu, J. Wang, B. Leng, and S. Zhang, "Unlocking the potential of reverse distillation for anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, 2025, pp. 5640–5648.

[42] L. P. J. Sträter, M. Salehi, E. Gavves, C. G. M. Snoek, and Y. M. Asano, "GeneralAD: Anomaly detection across domains by attending to distorted features," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 448–465.

[43] H. Feng, K. Song, W. Cui, Y. Zhang, and Y. Yan, "Cross position aggregation network for few-shot strip steel surface defect segmentation," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–10, 2023.

YAN-QIN NI is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, National Central University, Taoyuan City, Taiwan. Her research interests include semantic segmentation, anomaly detection, and smart manufacturing.

PEI-KAI HUANG (Graduate Student Member, IEEE) received the B.S. degree in software engineering from Fujian Normal University, Fujian, China, in 2017, the M.S. degree in computer science from National Central University, in 2019, and the Ph.D. degree in computer science from National Tsing Hua University, in 2025. As the first author, he has had 13 technical papers published at prestigious conferences, including CVPR, AAAI, BMVC, ICME, and ICIP, as well as in renowned journals, such as IEEE Transactions on Information Forensics and Security, *Pattern Recognition*, and *Information Sciences*. His research interests include multimedia content security, biometrics, and computer vision.

CHING-HAN YANG received the Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in 2018. He is currently a Senior Engineer with the Digital Transformation Research Institute, Institute for Information Industry, Taiwan. His research interests include intelligent manufacturing, cultural technology, and machine vision.

CHIN-CHUN CHANG (Member, IEEE) received the Ph.D. degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2000. He is currently a Professor with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, Taiwan. His research interests include computer vision, machine learning, and pattern recognition.

WEI-JEN WANG (Member, IEEE) received the Ph.D. degree in computer science from the Rensselaer Polytechnic Institute, New York, NY, USA, in 2006. He is currently a Professor with the Department of Computer Science and Information Engineering, National Central University, Taoyuan City, Taiwan. His research interests include cloud computing, edge computing, smart manufacturing, distributed programming, high-availability, and fault tolerance.

DERON LIANG received the Ph.D. degree in computer science from the University of Maryland at College Park, College Park, MD, USA, in 1992. He is currently a Professor and the Director of the Software Research Center, National Central University, Taoyuan City, Taiwan. His research interests include intelligent systems software, smart manufacturing, data analysis, software engineering, and information system security.