Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105
Volume **9**, Number **4**, November **2024**

Deep LBP-Enriched Real-time Segmentation for Hair Defect Detection

Xiaodong Wang, Jie Wang*, Zhiyao Xie, Fei Yan

College of Computer and Information Engineering
Xiamen University of Technology, Xiamen 361024, China
xdwangjsj@xmut.edu.cn, w1160404405@163.com, 914422782@qq.com, fyan@xmut.edu.cn

Haiyan Huang*

School of Computer Engineering
Guangzhou City University of Technology, Guangzhou 510800, China
huanghy2@gcu.edu.cn

Zhiqiang Zeng

College of Computer and Information Engineering
Xiamen University of Technology, Xiamen 361024, China
zqzeng@xmut.edu.cn

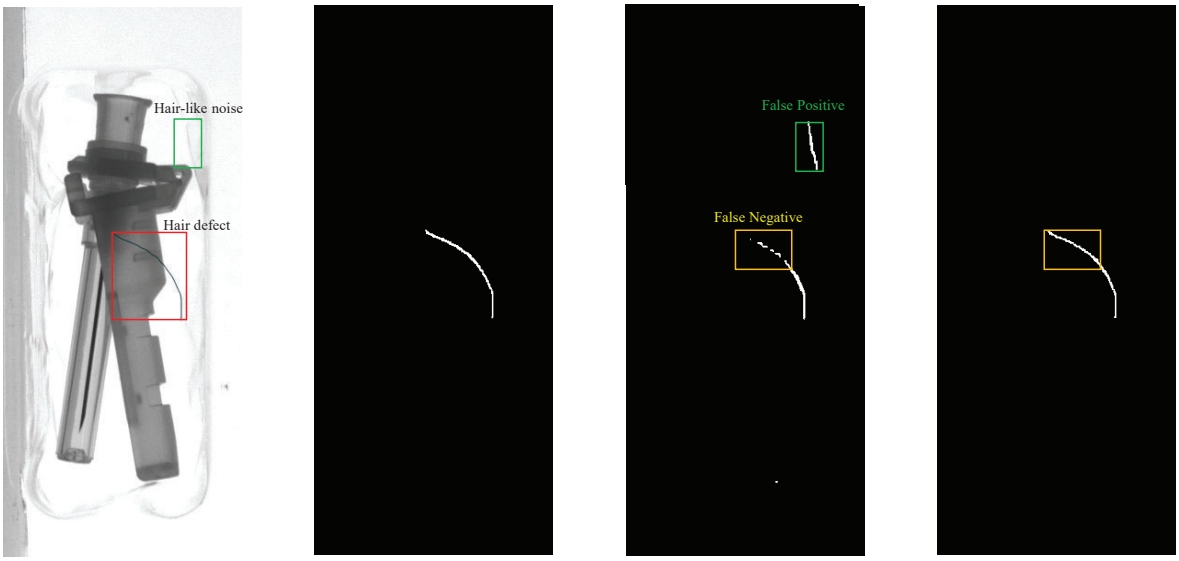
*Corresponding author: Jie Wang, Haiyan Huang

Received October 28, 2023, revised January 19, 2024, accepted April 17, 2024.

Abstract. *In the industrial production of medical syringes, existing Deep Semantic Segmentation (DSS) methods, which generally have numerous network parameters, face significant challenges in real-time hair defect detection due to hair's unique characteristics, including its irregular and thin structure. Moreover, potential hair overlapping with the syringe further complicates the detection process. In this case, conventional DSS methods are hard to explore the accurate low-level visuospatial information that is critical for detecting hair defects. Considering the wide applicability and effectiveness of the Local Binary Pattern (LBP) for detecting irregular and thin structural defects, we argue that LBP may help DSS models to extract more detailed hair features as a way to improve their performance in detecting hair defects on syringes. Inspired by this idea, this study proposes a Deep LBP-Enriched Real-time Segmentation (DLERS) method for hair defect detection, which maintains a lightweight network structure and leverages the LBP encoding mechanism to facilitate DSS models to extract more detailed edge features. Besides, to alleviate the influence of the hair-like noise and fragmentary edges, we propose employing a joint loss function that combines the Dice loss, BCE loss, and Edge loss to train our network. To evaluate the performance of DLERS, we conduct experiments on one real-world syringe dataset. The competitive results (*e.g.*, 85.36% MIoU and 149.1 FPS) prove the effectiveness of our method.*

Keywords: Deep learning, Local Binary Pattern, Real-time defect detection, Image segmentation

1. **Introduction.** Product defect detection is essential to industrial production and the key to controlling product quality. Generally, according to their sources, defects of products can be divided into two categories: internal or external. The internal defects, such as design flaws and material issues, originate from the product itself. The external defects refer to exogenous foreign-bodies caused by the external environment. These exogenous foreign-bodies occur due to the external factors of the production environment, which



(a) Input (b) Ground truth (c) Unet[8] (d) Ours

FIGURE 1. The input image with hair defect and its segmentation results. Given an input image (subfigure (a)) with hair defects, where red boxes represent hair defects and green boxes represent hair-like noise. Conventional DSS method [8] suffers from hair-missing issue (marked by the yellow box in subfigure (c)) and hair-like noise (marked by the green box in subfigure (c)). In contrast, our method can accurately extract defects (subfigure (d)), which is quite close to the ground truth (subfigure (b)).

will contaminate the product. Given the profound impact of these prevalent defects, it is essential to design contextually appropriate and efficacious detection techniques.

Hair defect is commonly identified as a type of exogenous defect because the bacteria carried by the oil on the hair's surface can seriously contaminate medical syringes. If healthcare workers use contaminated syringes to infuse patients, it may lead to inflammatory reactions and may even result in medical accidents in more severe cases. Therefore, to effectively eliminate this safety hazard, factories need an efficient and real-time hair defect detection method in industrial production. Conventionally, most existing hair defect detection programs are artificially conducted by workers, which, unfortunately, tend to be time-consuming, there by severely impeding industrial productivity. To increase production efficiency, experts have introduced classic machine vision technology to the field of defect detection [1] [2] [3] [4]. These technologies mainly use traditional Handcrafted Feature (HF) to detect defects. HFs are designed to capture defects directly from the pixel-level information and aligns with the characteristics observed by human vision. Generally, HFs in industrial defect detection rely on human-designed rules and require domain expert knowledge. While these features are typically interpretable, they often lack robustness, especially when dealing with complex, diverse, and noisy data.

With the rapid development of industrial manufacturing, medical syringe production lines have become more intelligent and efficient. Under this circumstance, conventional handcrafted-feature guided defect detection methods are vulnerable to interference and suffer from poor robustness, which limits their applications in medical syringe inspection. In recent years, due to their high accuracy, adaptability, and automatic feature extraction [5] [6] [7], deep semantic segmentation has been widely applied in defect detection. However, existing deep semantic segmentation methods encounter difficulties in achieving

satisfactory results for real-time industrial defect detection because of their substantial demand for computational resources and large amounts of training data. In addition, industrial cameras in real-world medical syringe production lines frequently capture subtle and hair-like internal wrinkles, constituting a critical challenge for defect detection on medical syringes. Complicating matters further, hairs may overlap with medical syringes, leading to a more intricate background in the captured syringe images. This complexity poses challenges for defect detection methods. In this case, it is difficult for conventional neural networks to detect defects effectively. Their segmentation results may suffer from false positive (the green box in Figure 1(c)) and false negative (the yellow box in Figure 1(c)), leading to inaccurate detection outcomes. Therefore, an efficient hair defect detection technology for medical syringes is imminent.

To solve the problems encountered in hair defects, we propose a lightweight method called Deep LBP-Enriched Real-time Segmentation (DLERS) method. This method employs an LBP feature encoding mechanism for DSS models, which offers advantages in extracting more detailed hair features and can be seamlessly adapted to various segmentation network frameworks. Specifically, our method is built on three observations: 1) In the hair defect detection, it is important for the segmentation network to utilize the low-level visuospatial information to explore fine-grained details of hair defects, such as edge features; 2) It is essential to mitigate the impact of hair-like noise and fragmentary edges in the syringe tray; 3) In industrial defect detection, deep segmentation models should be designed with efficient architectures to meet the real-time requirements of production lines. To leverage these findings, DLERS integrates the LBP feature encoding strategy to enhance the network's ability to capture essential features for segmentation. Additionally, DLERS maintains a lightweight network structure and utilizes a joint loss function that combines Dice loss, BCE loss, and Edge loss, to efficiently and robustly segment hair defects, even in the presences of hair-like noise. To summarize, the main contributions of this paper are listed as follows:

- We propose a lightweight Deep LBP-Enriched Real-time Segmentation (DLERS) method for real-time hair defect detection on syringes, which integrates the LBP feature encoding into a deep neural network, enhancing the detection performance by leveraging fine-grained hair information, *e.g.*, edge features.
- To better address the challenges posed by hair-like noise and fragmentary edges, we propose the utilization of a joint loss function that combines the Dice loss, BCE loss, and Edge loss. This joint loss function allows the network to learn distinctive features of hair defects, ensuring accurate defect segmentation.
- The competitive results on one real-world syringe dataset demonstrate the effectiveness of our method. Our method arrives at the best result, that is, 85.36% on MIoU and 149.1 Frames Per Second (FPS), which surpasses the compared methods over by 2.15% MIoU and more than four times faster in processing speed.

2. **Related work.** Our method closely relates to LBP-based defect detection and deep learning-based segmentation. Both of these methods have shown promising performance in defect detection. In this section, we briefly introduce these two techniques.

LBP-based defect detection: In the early stage of automatic defect detection, the traditional machine vision-based methods often utilize Local Binary Pattern (LBP) in images to detect surface defects. For instance, Li et al. [9] proposed an energy-based local binary patterns (ELBPs) rather than grayscale-based local binary patterns to detect structural defects. Liu et al. [10] proposed to extract the LBP features of tires to detect superficial crack-shaped defects. Song and Yan [11] proposed an adjacent evaluation completed local binary patterns (AECLBPs) algorithm. This method can detect surface

defects in hot rolled steel strips even under the influence of the feature variations of intra-class changes, illumination, and grayscale changes. Liu et al. [12] proposed a real-time detection algorithm for the steel surface defects. Similarly, Li et al. [13] proposed a texture description model for birch board crack. Utilizing the typical characteristics of complex geometry and texture distribution, Cao et al. [14] designed a nickel foam surface defect detection method.

Deep learning-based segmentation: Building upon success of deep neural networks in computer vision tasks, Masci et al. [15] proposed to apply deep learning technology to automatic visual surface defect detection. Their work shows that combining convolution neural networks and maximum pooling is superior to the handcrafted feature approach. Weimer et al. [16] proposed a neural network structure for industrial defect detection, which learned features from data and achieved a final average detection accuracy of 99.2%. To find subtle defects, Racki et al. [17] proposed a two-stage network based on a segmentation network that achieves excellent results in a weakly labeled anomaly dataset DAGM with different textured surfaces and different shapes. Tabernik et al. [18] also employed segmentation-based two-stage networks on the Kolektor surface-defect dataset and reached more advanced performance. In addition, many studies [19] [20] [22] [21] have found that low-level visuospatial features in the network can improve the performance of semantic segmentation network. On this basis, Wang et al. [23] proposes to solve the edge blurring problem by leveraging multilevel representations during segmentation. However, this simple addition or concatenation of multi-level features does not adequately capture low-level visuospatial features and can cause misalignment problems.

To address the aforementioned issue, Takikawa et al. [24] proposed a GSCNN network that reclassifies features into standard flow and shape flow. In this method, the Canny operator is used in the branch network to enhance the edge information in the shape flow to obtain better segmentation results. Luan et al. [25] proposed a Gabor convolution neural network (GaborCNN) by combining with traditional handcrafted features. They believe that Gabor can enhance the adaptability of deep learning features by handling changes in direction and scale. Zhao et al. [26] proposed a defect detection method combining GAN and LBP, which achieves excellent results on fabric images and the DAGM2007 dataset.

3. Methodology.

In this section, we describe our Deep LBP-Enriched Real-time Segmentation (DLERS) method in detail. First, we outline the overall structure of DLERS in Subsection 3.1. Then, we provide a comprehensive explanation of how our LBP feature encoding of DLERS is designed to extract more detailed features to enhance the DSS model for hair defect detection in Subsection 3.2. Finally, the joint loss function with Dice loss, BCE loss, and Edge loss is introduced in Subsection 3.3.

3.1. Network structure.

We propose an efficient and lightweight network for real-time detection of hair defects on medical syringes. The overall structure of DLERS is shown in Figure 2. We can see that DLERS contains three parts: base network, LBP-enriched Feature Encoding (LFE) modules, and semantic segmentation head (Seg_head). The architecture of our base network draws structural inspiration from the successful You Only Look Once (YOLO) [27] [28] [29] model, which has demonstrated excellent performance in various industrial production applications. The YOLO model family is considered one of the most advanced object detection models, known for its skip connection and feature extraction techniques that contribute to excellent detection performance. Besides, its lightweight character derives from the model's utilization of the bottleneck block. Based on these two characteristics, we build a lightweight base network that incorporates the bottleneck block along with a specially designed skip connection–the LFE modules,

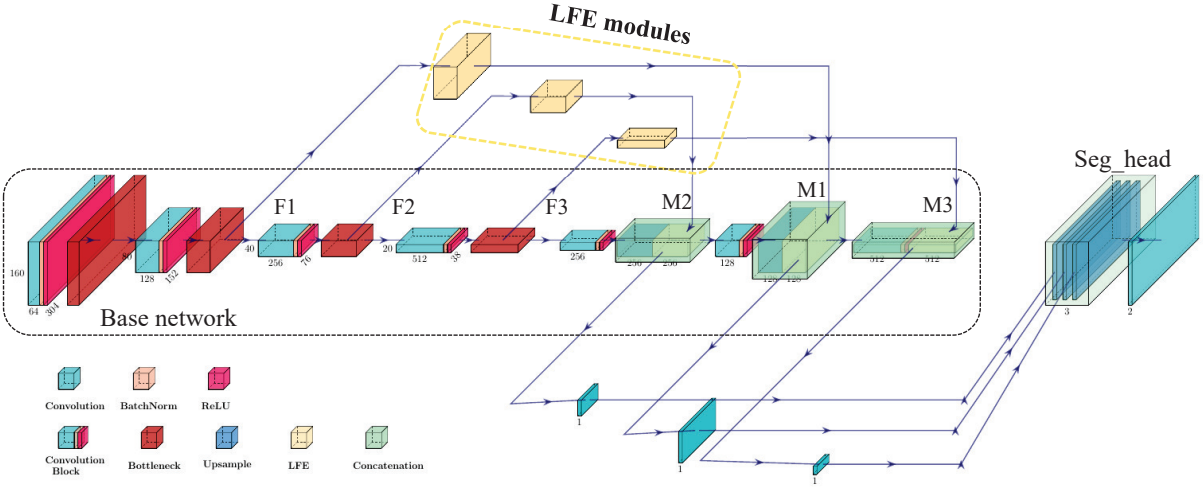


FIGURE 2. The network structure of our Deep LBP-Enriched Real-time Segmentation (DLERS) method. To extract detailed texture features of hair defects, we conduct LBP-enriched feature encoding modules (yellow dotted rectangle area) to augment the outputs of the base network with different feature scales (features at three different layers (F_1, F_2, and F_3)). Finally, the enriched texture features are fused and upsampled for semantic prediction in the segmentation head (Seg_head).

which will be described in the next paragraph. The base network consists of several key components, including the convolution block, bottleneck block, upsample layer, and concatenation layer. The bottleneck structure is adopted to reduce the number of channels by half using a 1×1 convolution, then double the number of channels through a 3×3 convolution to obtain high-level features.

Additionally, to extract rich low-level features, we construct LFE modules to explore fine-grained features of hairs defects. The objective of LFE modules is to learn a mapping function $f_\theta : F \to M$, which transforms the input feature F into a new feature space M using the parameters θ. In this paper, we use the LBP encoding technique to form the parameters θ, that is, f_K. Here, K refers to the kernel matrix generated from the LBP encoding. Detailed construction process of K will be discussed in Subsection 3.2. After the construction of LFE modules, we can apply it to the base network to enrich the low-level features. As shown in Figure 2, given the low-level features F_1, F_2, and F_3 from different layers, their LBP-enriched features M_1, M_2, and M_3 can be formulated as $F_1 \to M_1$, $F_2 \to M_2$, and $F_3 \to M_3$ by the mapping function f_K, respectively. It is worth noting that in the LFE modules, the kernel K generated from LBP is fixed. As shown in Figure 1, thanks to these LFE modules, our method can solve the challenges arising from the hair defect's irregular and thin structure and its potential overlap with the syringe.

To fuse features from different scales, we employ a 1×1 convolution block within the segmentation head of our network to condense the feature representation. After that, we upsample these fused features to the original image size. This step is crucial to align the fused features with the input image, permitting precise localization and accurate prediction of hair defects. Finally, with the fused and upsampled features, we introduce a convolution layer to generate the final prediction for hair defect detection.

3.2. **LBP-enriched feature encoding module.** The classical machine vision-based methods usually encode product surface defects via localized inhomogeneous texture patterns. Among these methods, the Local Binary Pattern (LBP) [30] is renowned for its superior resilience to noise and illumination variation, which has rendered it a highly efficient and widely used operator in many scenarios for extracting local texture features. Specifically, the traditional LBP operator works on patches of an image that typically has a size of $m \times m$ pixels, where m is usually an odd number such as 3, 5, *etc.* In LBP, neighboring pixels with higher intensity values than the central pixel are assigned 1 and 0 otherwise. In 2D space, each pixel I_{ij} located at (i, j) of an image can be encoded as a binary sequence $Q(i, j)$, which can be expressed as:

$$Q(i,j) = \left[S_1^{ij}, S_2^{ij}, \ldots, S_{4m-4}^{ij}\right], S_k^{ij} \in \{0,1\}, \tag{1}$$

where S_k^{ij} is the code of the kth neighboring pixel N_k^{ij}. Concretely, S_k^{ij} can be calculated as:

$$S_k^{ij} = \delta\left(N_k^{ij} - I_{ij}\right), \tag{2}$$

where $\delta(x) = 1$ if $x \geq 0$, $\delta(x) = 0$ otherwise. It should be noted that there are $4m - 4$ neighboring pixels in clockwise order for each I_{ij}. Formally, these neighboring pixels consist a sequence N^{ij} as follows:

$$N^{ij} = \left[N_1^{ij}, N_2^{ij}, \ldots, N_{4m-4}^{ij}\right]. \tag{3}$$

The above encoding process in Equation (3) of LBP, designed by domain experts, has been widely used in many computer vision tasks, such as texture classification and object recognition, for its compact representation and robustness to noise. This encoding strategy allows LBP to capture rich texture features in an image and effectively address complex problems, ultimately achieving high performance. Building on these findings, we believe that LBP can also enhance the performance of the DSS model in detecting hair defects on syringes. To tackle the challenge of hair defects with irregular and thin structures that commonly impede conventional DSS models' capacity to fully exploit low-level visuospatial information, we propose integrating LBP into the model. This addition enhances the DSS's texture feature extraction, thereby significantly improving its performance in detecting hair defects.

Given a low-level feature $X \in \mathbb{R}^{c \times h \times w}$ generated by the DSS model, we define a feature-enrichment function f_K to better capture the fine-grained texture for hair defects, where K is a predefined kernel constructed in accordance with the rules of LBP feature encoding. To utilize the prior knowledge in LBP, we can determine K according to the encoding framework in Equation (1). However, the binary sequence generated in Equation (1) for each pixel is not compatible with existing kernels in deep neural networks, making it hard to directly use it in the DSS model. To solve this problem, upon meticulous examination of Equation (1), we can observe that for each pixel there are $n = 2^{4m-4}$ possible encoding states according to the relative relations between the central pixel and its neighbors. Each encoding state p has $4m-4$ binary bits with predetermined values and represents a specific texture information. Based on these findings, we can build a kernel $\kappa \in \mathbb{R}^{m \times m}$ from p as follows:

$$\kappa_{i,j} = \begin{cases} \alpha & \text{if } i = j = \frac{m+1}{2} \\ p_{\phi(i,j)} & \text{otherwise} \end{cases} s.t.\ 1 \leq i, j \leq m, \tag{4}$$

where $\phi(i, j)$ is an index function, which calculates the index of the element located at (i, j) in κ starting from $\kappa_{1,1}$ in clockwise order. α is the central supplementary parameter of the kernel matrix. Drawing inspiration from traditional handcrafted features such as the Sobel and Laplacian operators, which maintain a central element with a value that

is the negative sum of the other elements in the kernel, we propose a similar approach. Concretely, we design our kernel such that its central parameter is exactly set to the negative sum of the remaining elements in the kernel, thereby enhancing its capacity to extract texture information from the data. Formally, α can be calculated as $\alpha = -\sum_{u=1}^{4m-4} p_u$. After applying all the possible encoding states to Equation (4), we can get distinct n kernels from LBP with an $m \times m$ image patch size.

Let $K = [\kappa^1, \kappa^2, \ldots, \kappa^n], K \in \mathbb{R}^{n \times m \times m}$ be the obtained kernel matrix with n kernels constructed by the kernel construction rules in Equation (4). We can apply these channels to each individual channel of low-level feature X. Suppose the enriched features across all n channels of X is T, the vth channel of T can be calculated as:

$$T_{v,i,j} = f_K(X) = \sum_{p=1}^{m} \sum_{q=1}^{m} \left(X_{v,i-p,j-q} \cdot K_{v,p,q} \right), \tag{5}$$

where $\cdot$ is the dot product. It is worth noting that Equation (5) works in a depth-wise separable manner, which is computational efficient. In this context, to ensure compatibility with subsequent layers, we utilize a 1×1 convolution W^b before feature enrichment to match the number of input channels and the number of output channels. Besides, after the feature enrichment, an additional 1×1 convolution W^a is introduced to combine the information from different channels. Finally, the merged texture M for the input data can be formulated as follows:

$$M = Conv(W^a, f_k(Conv(W^b, X))). \tag{6}$$

3.3. **Loss function.** Detecting hair accurately can be challenging due to its thin and delicate texture. To enhance the overall quality of hair detection, following the previous studies on edge detection [31], we propose to capture the distinct edges by exploring gradients associated with hair strands. Specifically, the Sobel operator, renowned for its simplicity and edge detection capability, is introduced to our hair detection framework. Mathematically, Sobel operator maps gradient changes in the 2-dimensional space with a 3×3 filter S_x in the horizontal direction and a 3×3 filter S_y in the vertical direction. Typically, the values of S_x and S_y are set as follows:

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \qquad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{7}$$

Suppose the predicted result and the ground truth of an input training image are denoted as P and Y, respectively. To highlight the distinguishable edges along with hairs, we hope that the hair boundaries in P and Y should be as close as possible. Based on this assumption, we propose to minimize the following Edge loss function:

$$L_{edge}(Y, P) = \mathbb{E}\left[\left| Y * S_y - P * S_y \right| + \left| Y * S_x - P * S_x \right| \right], \tag{8}$$

where $*$ is the convolution operation, $\mathbb{E}$ is the expectation operator, *i.e.*, mean expectation.

In addition, in the field of defect segmentation, pixel imbalance has been a long-standing problem, with researchers striving to address it for many years [32] [33]. Specifically, the small part of the defective region relative to the whole image poses difficulty for segmentation models. In this case, the segmentation model may prioritize the segmentation of non-defective pixels, neglecting defective pixels and reducing model performance. This issue is prevalent in hair detection due to the fine characteristics of hair strands.

Following the previous study [23], one way to solve this problem is to use the Dice loss function [34], which can allow the network to focus more on defective regions. The Dice

loss function can be expressed as:

$$L_{dice}(Y,P) = 1 - \frac{2\sum_{i=1}^{W}\sum_{j=1}^{H}(y_{ij}p_{ij}) + \epsilon}{\sum_{i=1}^{W}\sum_{j=1}^{H}(y_{ij}) + \sum_{i=1}^{W}\sum_{j=1}^{H}(p_{ij}) + \epsilon}, \tag{9}$$

where W and H denote the length and width of the input image. $y_{ij} \in \{0,1\}$ denotes the pixel at coordinate (i,j) in Y. ϵ is a small constant number used to smooth the loss function to avoid the problem of division by zero.

During the training process, it is common to encounter scenarios where the input data contains noise. In such cases, the presence of noise can cause the predicted result deviate from the ground truth, leading to potential inconsistencies or inaccuracies. It is important to note that while the Edge loss and the Dice loss have their respective advantages, they are not immune to the effects of noise. Take the Dice loss for an example. Consider a prediction $P = (1,0,0,0,0.2)$ with a ground truth $Y = (1,0,0,0,1)$ for a five-pixel input image, where 0, 1, and 0.2 correspond to defective, non-defective, and noisy pixels, respectively. Assuming $\epsilon = 0$ in Equation (9), it is expected that the interference caused by the noisy pixels should be mitigated in the subsequent training epoch. However, compared to the defect pixels, the noisy pixel has a minor impact on the denominator, making the gradient calculated by Dice loss ineffective in the next model update. To solve this problem, we leverage the BCE loss function to diminish the influence of noisy pixels as follows:

$$L_{bce}(Y,P) = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}\left(-y_{ij}log(p_{ij}) - (1-y_{ij})log(1-p_{ij})\right). \tag{10}$$

In a multi-tasking neural network [35], since each of the individual single-task loss functions may exhibit distinct significance, it becomes crucial to assign each of these tasks with a respective weight factor ω_i. In our study, instead of manually tuning ω_i, we take the adaptive learning approach suggested by Liebel and Kŏrne [36] to automatically adjust these weight factors. Specifically, We impose three learnable parameters: ω_1, ω_2, and ω_3, which are weight factors for the loss functions L_{bce}, L_{dice}, and L_{edge} , respectively. Finally, after combining these weight factors, our joint loss function can be expressed as:

$$L_{all}(Y,P) = \frac{1}{2\omega_1^2}L_{bce}(Y,P) + \frac{1}{2\omega_2^2}L_{dice}(Y,P) + \frac{1}{2\omega_3^2}L_{edge}(Y,P) + \sum_{i=1}^{3} ln(1+\omega_i^2), \tag{11}$$

where $\sum_{i=1}^{3} ln(1+\omega_i^2)$ is a regularization term for all weight factors.

4. **Experimental results and analysis.** In this section, we evaluate the segmentation performance of our proposed method for real-time hair defect detection with thorough ablation studies in segmentation accuracy and speed. Four widely-used segmentation models, *i.e.*, Unet [8], SegNet [37], SCUNet [38], and FSDNet [39], for defect detection, are introduced for comparison in this paper. To validate the effectiveness of our proposed LFE module, we also apply it to two recent defect detection networks (*i.e.*, SCUNet and FSDNet). Finally, we verify the necessity of the joint loss function.

4.1. **Implementation details.** In this paper, we collected data from real-world industrial medical syringe production lines to test the proposed method. Due to the low hair-defect rate, high-speed production process, and the high cost of human labor, it is difficult to collect sufficient training images. The collected dataset consists of 200 gray syringe images, with half being defective samples and other half non-defective. During the experiment, the whole dataset was divided into training, validation, and test sets with a ratio of 6 : 2 : 2. This resulted in 120 training images, 40 validation images, and 40 test

Algorithm 1 Our Deep LBP-enriched Feature Encoding

Input: The low-level feature $X \in \mathbb{R}^{c\times h\times w}$, the kernel size m, the before-enrichment convolution weight $W^b \in \mathbb{R}^{n\times c\times 1\times 1}$, the after-enrichment convolution weight $W^a \in \mathbb{R}^{c\times n\times 1\times 1}$.

Output: The rich texture M.

1: Construct the set of binary sequences $\{Q_1, \ldots, Q_n\}$ encoded by the neighboring pixels in Equation (1).
2: Initialize the Deep LBP-enriched kernel matrix $K = [\kappa^1, \kappa^2, \ldots, \kappa^n]$ by $\kappa^i = Q_i, \forall i, 1 \leq i \leq n$.
3: **for** $v = 1$ to n **do**
4: Change the number of channels of X from c to n using $\sum_{l=1}^{n} \left(W_{v,l}^b \cdot X_l\right)$.
5: **end for**
6: **for** $v = 1$ to n **do**
7: Calculate the enriched feature $T_{v,i,j}$ $(1 \leq i \leq h, 1 \leq j \leq w)$ at the coordinate (i, j) of the vth channel of T using Equation (5).
8: **end for**
9: **for** $v = 1$ to c **do**
10: Compute the vth merged LBP-enriched texture feature M_v as $M_v = \sum_{l=1}^{c} \left(W_{v,l}^a \cdot X_l\right)$.
11: **end for**
12: **return** M

images. Considering the scarcity of training data, training a high-performance DSS model can be challenging. Although the type of syringe and the image background remain relatively consistent in the industrial production of medical syringe products, the variations in image brightness and syringe position still significantly affect this production process and defect detection process. To address these factors in the production environment, following the recent studies in syringe defect detection [19], we extended the training set to 480 images using: random brightness adjustment, random contrast adjustment, and random rotation techniques. It should also be noted that existing pre-trained models were usually trained for application scenarios distinct from hair defect detection. Hence, in our experiments, all the compared networks were trained from scratch, randomly initialized using a standard normal distribution. Detailed experimental settings are shown in Table 1.

Table 1. Detailed parameter settings of all experiments.

Configurations	Value
Input image size	360×640
Epochs	400
Batch size	16
Learning rate	0.02
Learning rate strategy	Poly ($power = 0.9$)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Computer configuration	RTX2080TI × 4 with 11GB GPU memory
Framework	Pytorch

4.2. **Evaluation metrics.** Two widely-used metrics are imposed to evaluate our method and the compared methods: Mean Intersection over Union (MIoU) and Mean Pixel Accuracy (MPA). MIoU is calculated by first computing the Intersection over Union (IoU) for each predicted label and its true label across all classes separately. After that, MIoU is obtained by averaging these IoUs over all classes. MPA is another evaluation metric for semantic segmentation. Define Pixel Accuracy (PA) as the ratio of correctly predicted pixels in a category to the total number of pixels. Then, MPA is calculated by averaging the PAs of all classes. Suppose the number of categories is C, the formulation of MIoU and MPA can be expressed as:

$$\text{MIoU} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{FN_i + FP_i + TP_i}, \tag{12}$$

$$\text{MPA} = \frac{1}{C}\sum_{i=1}^{C}\frac{TP_i}{FP_i + TP_i}, \tag{13}$$

where TP_i, FP_i, and FN_i refer to the true positive, false positive, and false negative of category i, respectively. Both larger values of MIoU and MPA indicate better segmentation performance. In this study, our primary focus is on determining whether each individual pixel represents a defect or not, resulting in a binary classification task with the number of category ($C = 2$).

In addition, to measure the efficiency of the defect detection algorithms, we introduce the Frames Per Second (FPS) metric, which represents the number of images processed by the defect detection algorithm in one second.

Table 2. Comparison segmentation results and model size from different models for hair extraction on medical syringes.

Model	Loss function	MPA	MIoU	GFLOPs	Parameters (M)
UNet [8]	Dice loss	89.70%	82.89%	19.43	34.5
	Joint loss	90.14%	83.99%		
SegNet [37]	Dice loss	89.02%	80.42%	11.90	29.44
	Joint loss	88.54%	82.73%		
SCUNet [38]	Dice loss	85.86%	78.83%	11.28	20.26
	Joint loss	87.61 %	83.21%		
FSDNet [39]	Dice loss	89.82%	82.46%	61.11	17.54
	Joint loss	90.06 %	83.16%		
DLERS(Ours)	Dice loss	92.60%	83.40%	1.43	5.80
	Joint loss	**92.81**%	**85.36**%		

4.3. **Segmentation comparison results.** In this subsection, two standard networks that are commonly used for semantic segmentation, that is, UNet [8] and SegNet [37], and two recent published defect detection networks, *i.e.*, SCUNet [38] and FSDNet [39], are introduced for comparison.

As shown in Table 2, DLERS obtains the highest MIoU of 85.36%, outperforming the two standard segmentation networks (*i.e.*, Unet and SegNet) by 1.37% and 2.63%, respectively. Meanwhile, DLERS outperforms the two recently published detection networks, that is, SCUNet and FSDNet, by 2.15% and 2.20%, respectively. Besides, DLERS also achieves the highest MPA over other compared methods, indicating its superior performance in accurately segmenting hair defects on medical syringes. To investigate how our

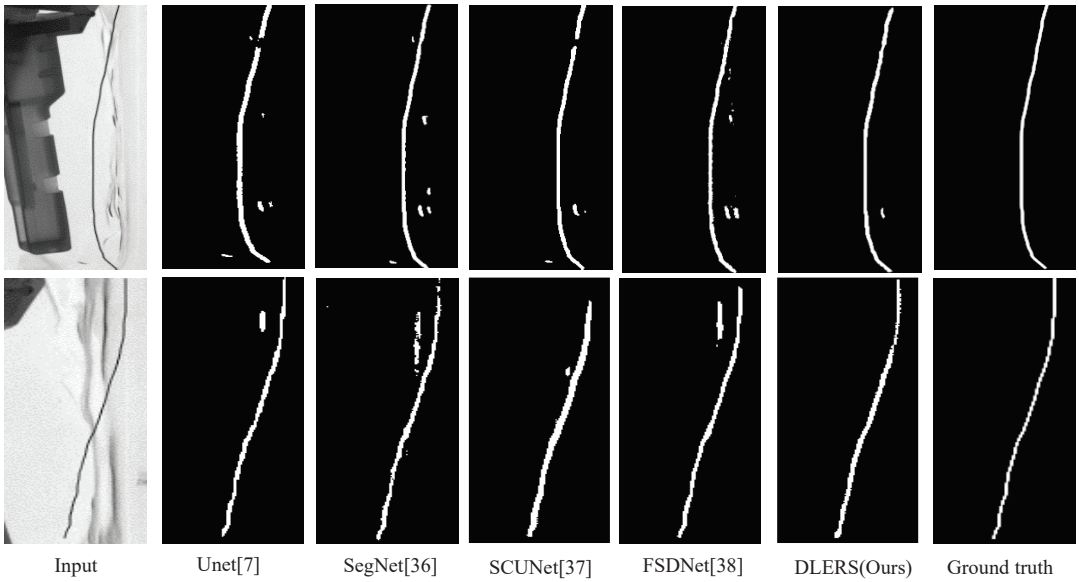


FIGURE 3. Visualization of hair extraction using different segmentation networks. We can see that most of the compared methods are easily effected by noise (the wrinkles in the syringe tray), resulting blurred or non-smooth segmentation results. In contrast, our method produces more accurate segmentation result, which is quite close the ground truth.

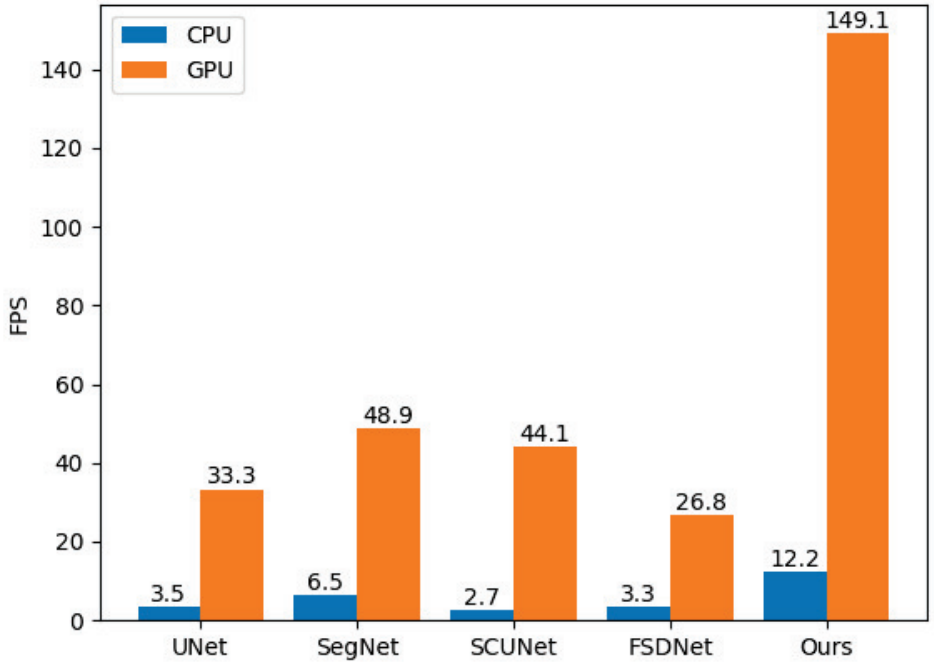


FIGURE 4. FPS comparison of different segmentation models on GPU (RTX2080TI) and CPU (Gold6242R).

method works in the hair detection scenario, we visualize its detection results in Figure 3 and compare it with other compared methods, *i.e.*, Unet, SegNet, SCUNet, and FSDNet. From the results, we can observe that neither of the four compared segmentation networks, that is, Unet, SegNet, SCUNet, and FSDNet, can distinguish the hair-like noise.

Although SCUNet performs better than SegNet and can remove some small-size or easily-distinguishable noise, it is still susceptible to noise that closely resembles hair defects. In contrast, our DLERS surpasses the compared methods by accurately identifying the hair defects with smooth segmentation, which hardly shows any gaps between the segmented pixels, positioning it as an efficient choice for real-time hair defect detection.

4.4. **Real-time performance analysis.** In Section 1, we have discussed the significance of real-time performance in industrial defect detection, highlighting its crucial role in ensuring the efficiency of production lines. In this subsection, we analyze and discuss the model complexity and the model size of all the compared networks. In addition, the real-time performance of the compared methods is also disucssed, focusing on their inference speed on different hardware platforms (*i.e.*, GPU and CPU).

As shown in Table 2, DLERS exhibits obvious advantages over SCUNet and FSDNet, especially in terms of network complexity and model size. Concretely, DLERS spends approximately 12% GFLOPs of SCUNet and SegNet, 7.4% GFLOPs of UNet, and 2.4% GFLOPs of FSDNet, with roughly 28.6% parameters of SCUNet, 19.7% parameters of SegNet, 16.8% parameters of UNet, and 33.1% parameters of FSDNet. These observations demonstrate that compared with other methods, our DLERS is more efficient for defect detection tasks. To further test how our method performs in real-time scenarios, we apply all the compared methods to different hardware platforms. Figure 4 shows the speed performance of the five methods in terms of frames processed per second (FPS) in Table 2. Notably, in the GPU environment, FSDNet shows the slowest processing speed, aligning with the GFLOPs in Table 2. The reason may be that FSDNet uses many 5×5 or even larger 15×15 convolution kernels throughout the network, which requires more computational resources. Thanks to the lightweight architecture and optimized design, our DLERS achieves the fastest segmentation speed (*i.e.*, 149.1 FPS on GPU), which is approximately three times faster than that of SegNet and SCUNet, and over four times faster than that of UNet and FSDNet. Similarly, in the CPU environment, our method still achieves the highest processing speed among the compared methods. To sum up, our DLERS offers a compelling solution for real-time defect detection, making it more suitable for high-speed and time-sensitive applications in industrial settings.

4.5. **Ablation experiments.** Our method utilizes the LFE modules to enrich the low-level features in deep learning models. In this subsection, we present visualizations and analyses of feature maps before and after the LFE modules to investigate their impact on low-level features. After that, to further demonstrate the effectiveness of our LFE modules, we also apply them to two recent defect detection networks, that is, SCUNet and FSDNet, and conduct comprehensive experiments to evaluate their performance. Finally, we evaluate the impact of our proposed joint loss function via detailed ablation studies.

The low-level visuospatial features visualization. We select DLERS's three low-level features from F_1, F_2, and F_3 (shown in Figure 2) for illustration. The visualization results are shown in Figure 5. We can see that before the LFE modules (first row of each subfigure), compared to other objects, such as the syringes, the low-level visuospatial information of the hair in the feature maps is not obvious, and the extracted hair features occupy only a small part of each feature map. Moreover, only certain easily observable attributes of these low-level features, such as boundaries, are identified. After the LFE modules (the second row of each subfigure), we can see that hair's extracted texture features are enhanced, leading to clearer and more distinct feature maps. Meanwhile, our LFE modules can significantly reduce the impact of noise and achieve more accurate hair defect segmentation results.

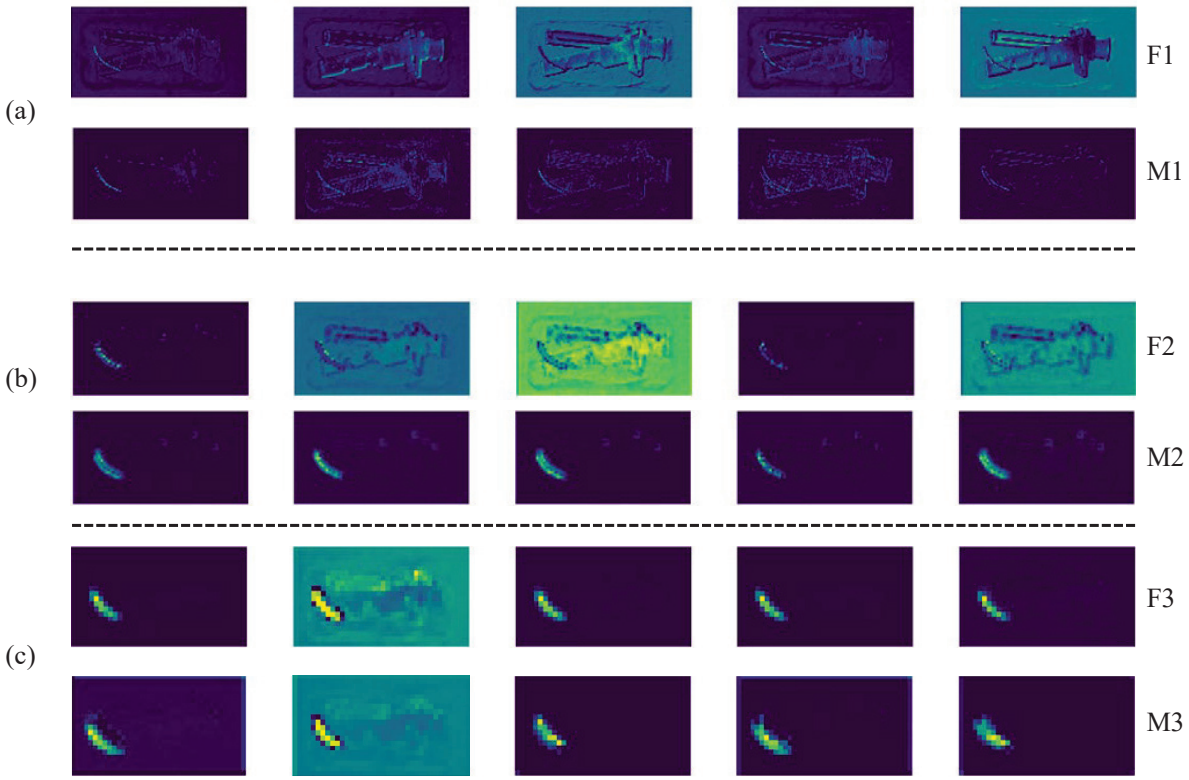


FIGURE 5. Partial visualization of low-level visuospatial features at different layers before (F1, F2, and F3) and after (M1, M2, and M2) our LFE modules on our medical syringe dataset.

Performance variation of our LFE modules with different defect detection networks. We select two recent networks, that is, SCUNet and FSDNet, in industrial production for this experiment, and apply three LFE modules to them. These three LFE modules take inputs from three specific layers. For SCUNet, these layers correspond to the second, third, and fourth layers, respectively. For FSDNet, the three layers are the second, third, and fourth layers, respectively. In SCUNet, we add the LFE modules directly to the copy operation of its U-structure. In FSDNet, the three outputs of the LFE modules are concatenated and subsequently serve as the input to the deconvolution layer. Table 3 shows the segmentation results after adding the LFE modules into SCUNet and FSDNet. For better presentation, we name these two newly created networks as SCUNet_LFE (the third row) and FSDNet_LFE (the fifth row). Clearly, we can see that our LFE modules promote significant improvements for these two networks in segmentation performance. For example, after adding the LFE modules, we achieves 2.34% and 1.80% improvements on MIoU over SCUNet and FSDNet, respectively.

Moreover, we also visualize the hair segmentation heatmaps of the methods listed in Table 3, as depicted in Figure 6. In this experiment, two representative input images are selected for comparison. In the first image, the hair partially overlaps with the medical syringe, whereas in the second image, there depicts a complete overlap. According to these visualized heatmaps, it can be found that SCUNet and FSDNet are partially able or completely unable to detect hairs that overlapped with the syringe. However, after adding our LFE modules, they can get improved segmentation heatmaps even when the

TABLE 3. Quantitative comparison of hair segmentation performance after applying our LFE modules to SCUNet and FSDNet.

Base network	LFE modules	MIoU(%)
SCUNet [38]	✗	83.21
SCUNet_LFE	✓	85.56
FSDNet [39]	✗	83.16
FSDNet_LFE	✓	84.96

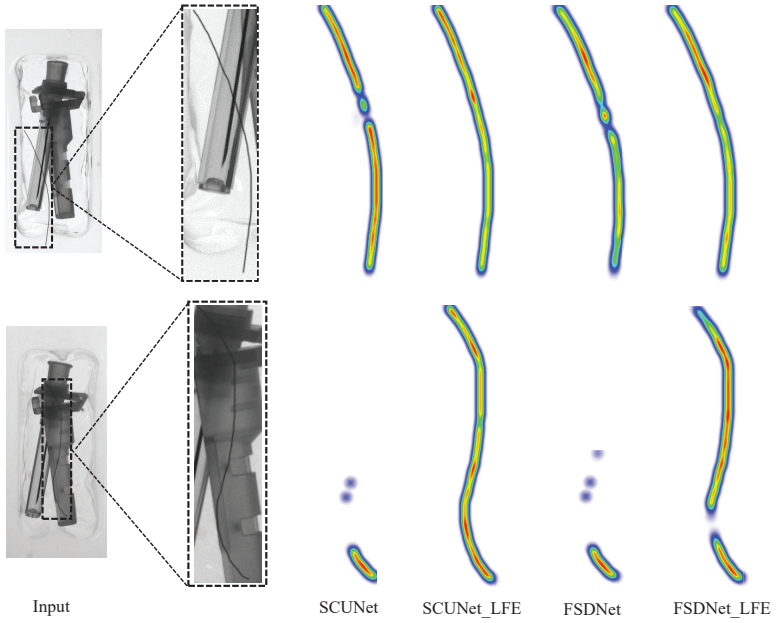


FIGURE 6. Visualization of hair segmentation heatmap using recent defect detection networks. These four segmentation results (SCUNet, SCUNet_LFE, FSDNet, and FSDNet_LFE) from left to right correspond to the four cases in Table 3 (from top to bottom).

hairs and syringes overlap entirely. These experimental findings again demonstrate the effectiveness of our LFE modules.

TABLE 4. DLERS's performance variations under different loss functions.

Model	Dice Loss	BCE Loss	Edge Loss	MIoU(%)
DLERS	✓			83.40
		✓		83.85
	✓	✓		84.26
	✓		✓	84.33
		✓	✓	83.88
	✓	✓	✓	85.36

Performance variation with different loss functions. To evaluate the validity of our joint loss function, we analyse the performance variations of DLERS using different

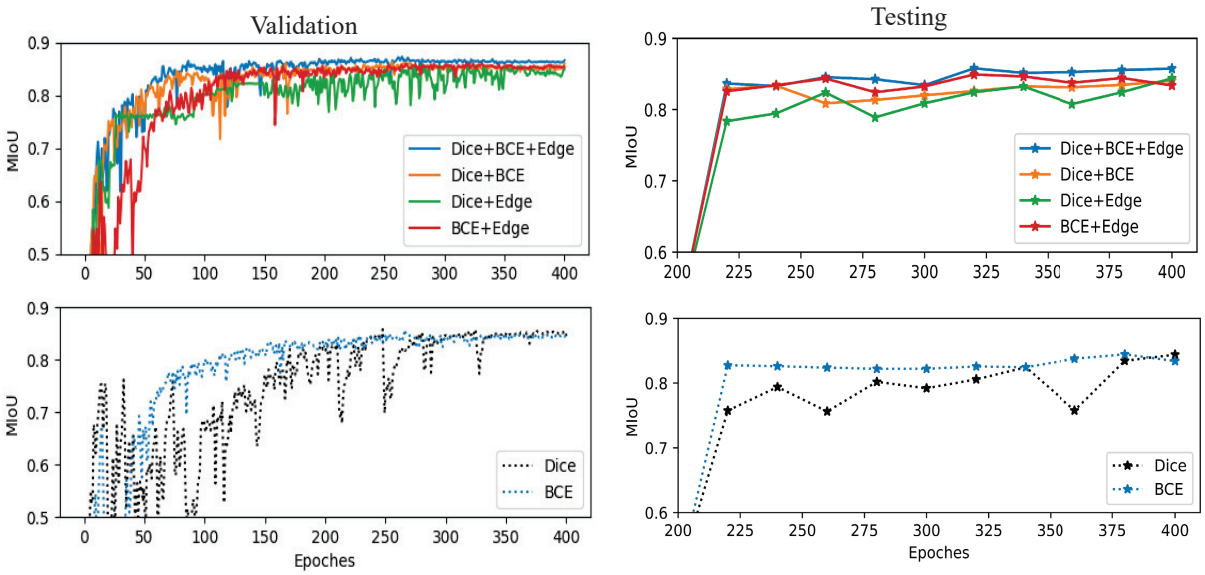


FIGURE 7. Performance variation of our DLERS network with different loss functions over 400 training epochs. The left part is the result on the validation sets and the right part is the result on the test sets.

loss functions and visualize the testing and validation MIoU at each training epoch. Table 4 shows the segmentation results of DLERS with different combinations of individual loss functions within our joint loss framework, which incorporates Dice loss, BCE loss, and Edge loss at different instances. When using Dice loss or BCE loss alone, the MIoU was 83.40% and 83.85%, respectively. After applying Dice+BCE, Dice+Edge, or BCE+Edge, MIoU is increased to 84.26%, 84.33% and 83.88%, respectively. The results show that employing any dual combinations of these loss functions can improve the hair defect segmentation performance of DLERS. Moreover, when we combine the three loss functions, DLERS achieves the best MIoU, *i.e.*, 85.36%. To explore the effect of each loss function on DLERS during the training process, we visualize DLERS's performance variations on the validation and test sets. Figure 7 shows our DLERS network optimization results under different combinations of loss functions (top subfigure) or a single loss function (bottom subfigure). As expected, the BCE loss can improve the network's performance and stability by reducing the impact of noise. This is evident from the fluctuating network performance in the absence of the BCE loss. This is indicated by the black dotted line in the bottom subfigure and the green solid line in the top subfigure, both of which are susceptible to noise interference. Additionally, the performance on the test set aligns consistently with that of the validation set. Finally, we can find that our DLERS network achieves the best performance when all three loss functions are utilized simultaneously, highlighting the significance of their combined usage. It is worth noting that the network is not trained solely with Edge loss due to its inherent limitations. Edge loss is designed to primarily constrain object edges along the gradient direction, making it challenging for pixel-level classification. In this case, relying solely on Edge loss may fail to provide sufficient guidance for the network to accurately learn discriminative features.

5. **Conclusions and Future Work.** In this paper, we propose a lightweight DLERS (Deep LBP-Enriched Real-time Segmentation) method for real-time hair defect detection. The proposed method consists of a base network and an LBP-enhanced feature encoding module. The base network, characterized by its lightweight architecture, mainly

undertakes the task of effectively extracting deep features. This LBP-enhanced feature encoding technique tackles the challenge of distinguishing hairs from fragmentary edges, by leveraging the strength of the Local Binary Patterns (LBP) in detecting fine-grained features of hair defects. By constructing and incorporating the LBP-enriched Feature Encoding (LFE) technique, we enhance the deep semantic segmentation model's ability to capture more comprehensive low-level visuospatial features. Additionally, we introduce a joint loss function that addresses the challenges associated with hair detection and alleviate the the impact of hair-like noise. Extensive experiments and ablation studies on a real-world syringe dataset demonstrate the effectiveness of our proposed methods.

In view of potential deployment on devices with limited computational resources, we acknowledge the importance of pursuing further optimization to enhance its efficiency and adaptability. As part of our future work, to make our model more compatible with these devices, we plan to use pruning techniques [40] to make our model slimmer.

Acknowledgment. This paper was supported by National Natural Science Foundation of Fujian Province (Grant Nos.2023J011428, 2022J011236, 2023J011426), Natural Science Foundation of Xiamen (3502Z20227073), Fuxiaquan National Independent Innovation Demonstration Zone collaborative innovation platform project (3502ZCQXT2021009), the University Industry Research Fund of Xiamen (Grant No.2022CXY0416).

REFERENCES

[1] H. Oh, N. W. Garrick, and L. E. Achenie, "Segmentation algorithm using iterative clipping for processing noisy pavement images," *Imaging Technologies: Techniques and Applications in Civil Engineering. Second International ConferenceEngineering Foundation; and Imaging Technologies Committee of the Technical Council on Computer Practices, American Society of Civil Engineer*, 1998.

[2] Y. Hu and C. Zhao, "A novel LBP based methods for pavement crack detection," *Journal of Pattern Recognition Research*, vol. 5, no. 1, pp. 140–147, 2010.

[3] X. Z. Yang, G. K. Pang, and N. H. C. Yung, "Discriminative fabric defect detection using adaptive wavelets," *Optical Engineering*, vol. 41, no. 12, pp. 3116–3126, 2002.

[4] K. Wang, C.-M. Chen, M. S. Hossain, G. Muhammad, S. Kumar, and S. Kumari, "Transfer reinforcement learning-based road object detection in next generation IoT domain," *Computer Networks*, vol. 193, 108078, 2021.

[5] F.-G. Tan, F.-M. Zhou, L.-S. Liu, J.-X. Li, K. Luo, Y. Ma, L.-X. Su, H.-L. Lin, and Z.-G. He, "Detection of wrong components in patch component based on transfer learning," *Journal of Network Intelligence*, vol. 5, no. 1, pp. 1–9, 2020.

[6] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li , "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, pp. 1–15, 2019.

[7] J.-L. Cui, Z. Wang, L.-H. Ma, Z.-M. Lu, and H.-L. Li, "An Intelligent Video Surveillance System Based on Object Recognition and Abnormal Behaviour Detection," *Journal of Network Intelligence*, vol. 6, pp. 3, 2021.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, 2015.

[9] W. Li, W. Xue, and L. Chen, "Intelligent detection of defects of yarn-dyed fabrics by energy-based local binary patterns," *Textile Research Journal*, vol. 82, no. 19, pp. 1960–1972, 2012.

[10] H. Liu, X. Jia, C. Su, H. Yang, and C. Li, "Tire appearance defect detection method via combining HOG and LBP features," *Frontiers in Physics*, vol. 10, 1099261, 2023.

[11] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Applied Surface Science*, vol. 285, pp. 858–864, 2013.

[12] Y. Liu, K. Xu, and J. Xu, "An improved MB-LBP defect recognition approach for the surface of steel plates," *Applied Sciences*, vol. 9, 4222, 2019.

[13] S. Li, D. Li, and W. Yuan, "Wood defect classification based on two-dimensional histogram constituted by LBP and local binary differential excitation pattern," *IEEE Access*, vol. 7, pp. 145829–145842, 2019.

[14] B.-F. Cao, J.-Q. Li, and N.-S. Qiao, "Nickel foam surface defect detection based on spatial-frequency multi-scale MB-LBP," *Soft Computing*, vol. 8, no. 8, pp. 5949–5957, 2020.

[15] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel defect classification with max-pooling convolutional neural networks," *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, 2012.

[16] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016.

[17] D. Racki, D. Tomazevic, and D. Skocaj, "A compact convolutional neural network for textured surface anomaly detection," *in Winter Conference on Applications of Computer Cision (WACV)*. IEEE, 2018, pp. 1331–1339.

[18] D. Tabernik, and S. Šela and 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.

[19] X. Wang, X. Xu, Y. Wang, P. Wu, F. Yan, and Z. Zeng, "A robust defect detection method for syringe scale without positive samples," *The Visual Computer*, vol. 39, no. 11, pp. 5451–5467, 2023.

[20] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 1925–1934.

[21] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," *in Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 325–341.

[22] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *in Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 801–818.

[23] X. Wang, Y. Wang, X. Xu, F. Yan, and Z. Zeng, "Two-stage deep neural network with joint loss and multi-level representations for defect detection," *Journal of Electronic Imaging*, vol. 31, no. 6, pp. 063060–063060, 2022.

[24] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," *in Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, 2019, pp. 5229–5238.

[25] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor convolutional networks," *IEEE Transactions on Image Processing*, vol. 27,no. 9, pp. 4357–4366, 2018.

[26] Z. Zhao, B. Li, R. Dong, and P. Zhao, "A surface defect detection method based on positive samples," *in PRICAI 2018: Trends in Artificial Intelligence: 15th Pacific Rim International Conference on Artificial Intelligence, Nanjing, China, August 28–31, 2018, Proceedings, Part II 15*. Springer, 2018, pp. 473–481.

[27] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," *arXiv preprint arXiv:2305.09972*, 2023.

[28] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2023, pp. 7464–7475.

[29] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie et al., "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[30] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[31] L. Liu, X. Jia, J. Liu, and Q. Tian, "Joint demosaicing and denoising with self guidance," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 2240–2249.

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *in Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2017, pp. 2980–2988.

[33] G. Cheng, J. Cheng, M. Luo, L. He, Y. Tian, and R. Wang, "Effective and efficient multitask learning for brain tumor segmentation," *Journal of Real-Time Image Processing*, vol. 17, pp. 1951–1960, 2020.

[34] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *in 2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 565–571.

[35] A. Kendall, Y. Gal, and R. Cipo, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.IEEE, 2018, pp. 7482–7491.

[36] L. Liebel and M. Kŏrne, "Auxiliary tasks in multi-task learning," *arXiv preprint arXiv:1805.06334*, 2018.

[37] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[38] L. Cheng, J. Yi, A. Chen, and Y. Zhang, "Fabric defect detection based on separate convolutional UNet," *Multimedia Tools and Applications*, vol. 82, no. 2, pp. 3101–3122, 2023.

[39] Y. Huang, J. Jing, and Z. Wang, "Fabric defect segmentation method based on deep learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[40] X. Wang, Z. Zheng, Y. He, F. Yan, Z. Zeng, and Y. Yang, "Soft person reidentification network pruning via blockwise adjacent filter decaying," *IEEE Transactions on Cybernetics*, vol. 52, pp. 13293–13307, 2021.