Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.075203

ARTICLE

MSC-DeepLabV3+: A Segmentation Model for Slender Fabric Roll Seam Detection

Weimin Shi[1,*], **Kuntao Lv**[1], **Chang Xuan**[1] **and Ji Wu**[2]

[1]School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou, 310018, China
[2]School of Applied Math & Computational Science, Duke Kunshan University, Kunshan, 215316, China

*Corresponding Author: Weimin Shi. Email: swm@zstu.edu.cn

Received: 27 October 2025; Accepted: 09 December 2025; Published: 12 March 2026

ABSTRACT: The application of deep learning in fabric defect detection has become increasingly widespread. To address false positives and false negatives in fabric roll seam detection, and to improve automation efficiency and product quality, we propose the Multi-scale Context DeepLabV3+ (MSC-DeepLabV3+), a semantic segmentation network designed for fabric roll seam detection, based on DeepLabV3+. The model improvements include enhancing the backbone performance through optimization of the UIB-MobileNetV2 network; designing the Dynamic Atrous and Sliding-window Fusion (DASF) module to improve adaptability to multi-scale seam structures with dynamic dilation rates and a sliding-window mechanism; and utilizing the Progressive Low-level Feature Fusion (PLFF) module to progressively restore seam boundary details via shallow feature fusion. Additionally, an enhanced 3-SE attention mechanism is employed, replacing the direct concatenation operation. Experimental results show that MSC-DeepLabV3+ outperforms classical and recent segmentation models. Compared to DeepLabV3+ with an Xception backbone, MSC-DeepLabV3+ achieves a mean intersection over union (mIoU) of 92.30% and the boundary F-score (BF) of 92.54%, representing improvements of 3.04% and 3.14%, respectively. Moreover, the model complexity is significantly reduced, with the model parameters (params) decreasing to 3.44M and Frames Per Second (FPS) increasing from 101 to 273, demonstrating its potential for deployment in resource-constrained industrial scenarios.

KEYWORDS: Fabric roll seam detection; semantic segmentation; deep learning; lightweight network; multi-scale feature extraction; improved attention mechanism

1 Introduction

During automated weaving and cutting, the need for continuous fabric splicing and hemming inevitably results in the formation of linear structures in fabric rolls, such as hem and joint seams, which are considered potential quality defects [1]. The accuracy of joint seam positioning directly affects the precision of subsequent cutting and automatic obstacle avoidance. Inaccurate positioning causes dimensional deviations and fabric wastage. Hem seams reflect both the manufacturing process and the appearance quality of the finished product, while detection errors lower the product pass rate. Therefore, accurate joint seam detection is crucial for ensuring product quality and production efficiency, making it a key challenge in textile manufacturing [2]. Fabric roll seam detection still mainly relies on manual inspection, which is inefficient, error-prone, and costly [3]. Some studies have used capacitive or thickness sensors for seam detection, but these sensors can only detect large-scale structural variations, which have poor sensitivity to subtle seam features and fail to provide accurate boundary information.

Copyright © 2026 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As computer vision advances, machine vision-based methods have been increasingly applied to fabric defect inspection [4]. Early methods primarily relied on handcrafted statistical and frequency-domain features, such as Local Binary Patterns (LBP) [5,6], Markov Random Fields (MRF) [7], Fourier Transform (FT) [8], and Wavelet Transform (WT) [9], combined with rule-based or traditional learning algorithms for defect recognition. Some studies also utilized thermal imaging by analyzing surface temperature gradients to identify hidden defects [10,11]. Although these methods improved detection accuracy compared to manual inspection, they rely on manually designed features and are sensitive to texture variations, lighting inconsistencies, and noise. As a result, traditional algorithms struggle to distinguish slender structures like seam heads, which often have blurred boundaries and textures similar to the background [12].

With the rapid development of deep learning, convolutional neural networks (CNNs) have achieved significant progress in feature representation and detection accuracy. Object detection models are widely used in industrial quality inspection due to their high inference speed and localization precision. Xu et al. proposed the YOLOv8-FPCA model, which improves sewing defect detection by enhancing the small object detection head and integrating an attention mechanism [13]. Guo et al. integrated the Atrous Spatial Pyramid Pooling (ASPP) module and channel attention mechanism into the YOLOv5 framework to improve resistance to interference in fabric defect detection [14]. However, object detection frameworks rely on rectangular bounding boxes for region localization [15], which has notable limitations in detecting slender fabric roll seams. These boxes often fail to capture the linear trajectory and local bending of the seams, leading to boundary loss or truncation. Additionally, object detection methods usually provide only rough outlines, lacking precise contour details. This makes the output unsuitable for tasks like detailed analysis and automatic cropping. In contrast, semantic segmentation models process pixels individually, fully reconstructing seam morphology, preserving boundary continuity, and providing more precise contours. As a result, semantic segmentation is more effective for detecting slender structures [16].

Semantic segmentation predicts the class of each pixel in an image, enabling precise classification and spatial localization [17]. This field has evolved from FCN's pixel-level prediction [18] to UNet's skip-connection architecture [19], PSPNet's multi-scale contextual fusion [20], and DeepLabV3+'s dilated convolutions with encoder-decoder fusion mechanisms [21], continuously improving segmentation accuracy and boundary preservation. However, many existing models for semantic segmentation are designed for general applications. When applied to textile detection, these models often struggle with complex textures and lighting variations. Guo et al. introduced adaptive channel and category weighting strategies for PSPNet to address sample imbalance and improve recognition accuracy in complex backgrounds, but its ability to generalize across different scenarios remains limited [22]. Hu et al. proposed a seam break detection method based on UNet, achieving precise localization, yet it is still susceptible to interference from variable lighting and complex textures [23].

Segmenting slender objects often faces challenges such as discontinuous shapes and background interference, particularly in complex textured backgrounds. Research indicates that DeepLabV3+, incorporating dilated convolutions and ASPP modules, effectively captures long-range contextual information. It achieves optimal performance in segmenting slender objects on datasets like Cityscapes and PASCAL VOC 2012 [24]. Nguyen applied DeepLabV3+ to sewing thread defect segmentation, combining it with an EfficientNet-B1 encoder to enhance detection accuracy, although boundary discontinuities persisted in the predicted regions [25]. Bai and Jing Mobile-DeepLab integrates a lightweight network with the DeepLab framework, though its generalization capability is limited by small sample sizes [26]. Liu et al.'s FP-DeepLab achieves fabric defect detection by enhancing the backbone network and multi-scale features, but remains limited to specific textile scenarios [27].

Existing models based on DeepLabV3+ and its variants mainly focus on improving feature extraction by enhancing the backbone network. However, they still rely on the ASPP module, which uses fixed dilation rates. This limitation leads to suboptimal performance in handling multi-scale objects and fine details, especially in tasks with complex structures like fabric roll seams. In seam detection, these limitations become more pronounced, as seams vary in scale, orientation, and texture, requiring more flexible and adaptive feature processing.

To address these limitations, we propose MSC-DeepLabV3+, a targeted improvement model based on DeepLabV3+, offering an end-to-end lightweight semantic segmentation solution. Building upon the encoder-decoder structure of DeepLabV3+, our model addresses the challenges of parameter efficiency and detail extraction, leading to improved fabric roll seam segmentation accuracy while reducing the parameter count and computational complexity. The main contributions are as follows:

1. Different from traditional defect classification and target localization algorithms, we propose the MSC-DeepLabV3+ based on DeepLabV3+, which optimizes multi-scale adaptability for fabric roll seam segmentation, addressing the limitations of traditional methods and achieving precise segmentation of slender seams.
2. The UIB-MobileNetV2 backbone was designed to optimize the feature extraction network, the DASF module was proposed to address the limitations of fixed dilation rates, and the PLFF module was introduced to progressively restore seam boundary details. By integrating these enhancements, the model achieves improved segmentation accuracy.
3. The model strikes a balance between reduced parameters and computational complexity, enabling it to effectively support real-time deployment on resource-constrained industrial platforms, while preserving high segmentation accuracy.

2 Proposed Model

DeepLabV3+ adopts an encoder-decoder architecture. In the encoder, the Xception backbone generates shallow and deep feature maps at different resolutions for feature extraction. The ASPP module employs dilated convolutions to capture multi-scale features. In the decoder, bilinear upsampling is applied to align and fuse shallow and deep feature maps, generating the final segmentation prediction. Based on DeepLabV3+, we propose MSC-DeepLabV3+, as shown in Fig. 1. Our model utilizes a lightweight UIB-MobileNetV2 backbone for feature extraction from fabric seam images. The DASF module improves the model's adaptability to multi-scale seam structures by dynamically adjusting dilation rates. In the decoder, the PLFF module integrates multi-level features, enhancing boundary details and maintaining spatial continuity for precise segmentation.

2.1 Replacement of the Backbone Feature Extraction Network

Seam defects exhibit varied and complex textures. The Xception backbone is computationally expensive, making it unsuitable for real-time detection. While MobileNetV2 is known for its lightweight design, its fixed Inverted Bottleneck (IB) block limits multi-scale perception and boundary capture.

To address this issue, we introduce a configurable Universal Inverted Bottleneck (UIB) block, which replaces the traditional IB block in MobileNetV2. The UIB block enhances feature representation and multi-scale modeling capabilities while maintaining a low parameter count. As shown in Fig. 2, the UIB block optimizes the traditional IB block by introducing flexible components such as expansion layers, projection layers, and optional depthwise convolutions, supporting four substructures. These components allow for task-specific adjustments in kernel size, stride, and position. By integrating the UIB block into the feature layers of MobileNetV2, we first expand the feature dimensions with additional depth, followed

by lightweight convolution using ConvNext for multi-scale feature extraction. Finally, features are refined using the traditional IB block, resulting in an efficient backbone designed for seam head segmentation—UIB-MobileNetV2. The network structure of UIB-MobileNetV2 is detailed in Table 1. This design preserves the model's lightweight nature while also enhancing multi-scale feature extraction and texture representation, thereby improving the feature expression for seam segmentation.

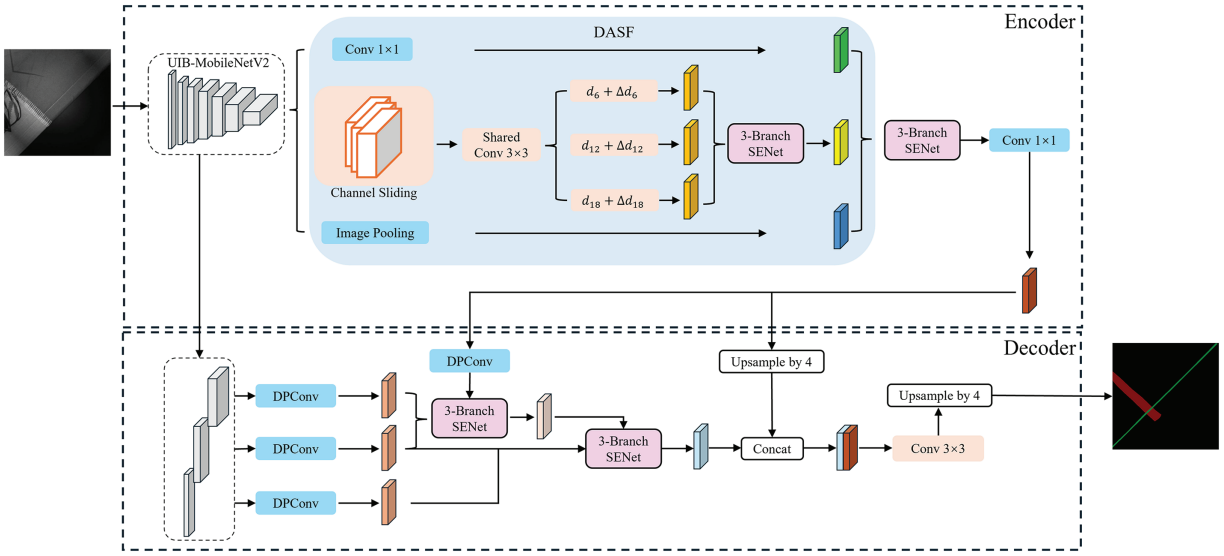

Figure 1: MSC-DeepLabV3+ model

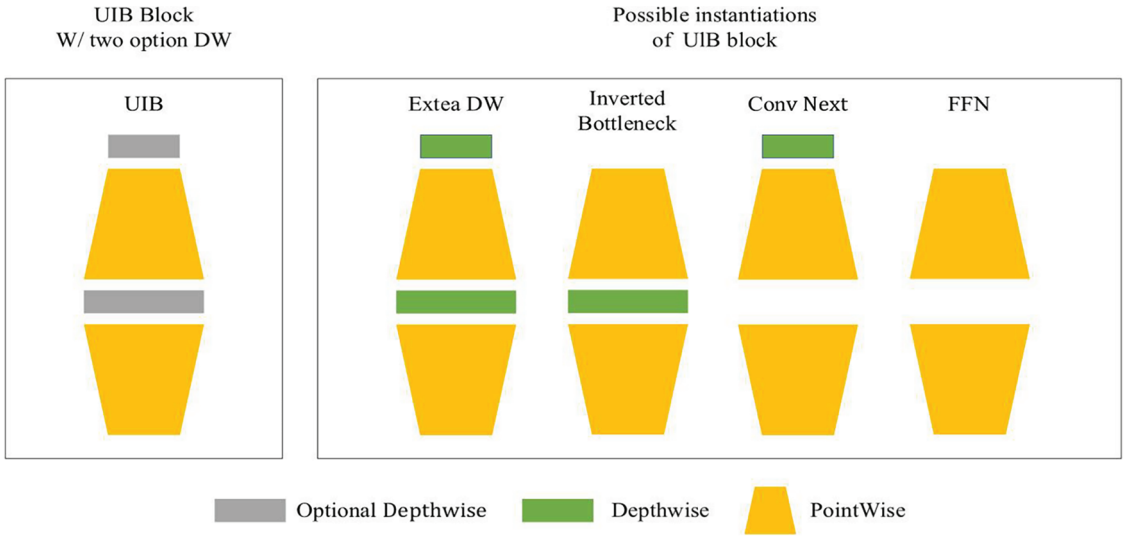

Figure 2: Universal inverted bottleneck (UIB) blocks

Table 1: UIB-MobileNetV2 network structure

Layer	Block	Input	Initial DW	Middle DW	Stride	Exp	Output
Layer0	Conv	512 × 512 × 3	3 × 3 Conv	–	2	–	256 × 256 × 32
Layer1	Conv	256 × 256 × 32	3 × 3 Conv	–	1	–	256 × 256 × 16
Layer2	Extra DW	256 × 256 × 16	3 × 3 DW	3 × 3 DW	2	3	128 × 128 × 24
	ConvNext	128 × 128 × 24	3 × 3 DW	–	1	2	128 × 128 × 24
Layer3	Extra DW	128 × 128 × 24	5 × 5 DW	5 × 5 DW	2	3	64 × 64 × 32
	ConvNext	64 × 64 × 32	5 × 5 DW	–	1	2	64 × 64 × 32
	IB	64 × 64 × 32	–	3 × 3 DW	1	8	64 × 64 × 32
Layer4	Extra DW	64 × 64 × 32	3 × 3 DW	3 × 3 DW	2	2	32 × 32 × 64
	ConvNext	32 × 32 × 64	3 × 3 DW	–	1	3	32 × 32 × 64
	IB	32 × 32 × 64	–	3 × 3 DW	1	6	32 × 32 × 64
	IB	32 × 32 × 64	–	3 × 3 DW	1	6	32 × 32 × 64
Layer5	Extra DW	32 × 32 × 64	3 × 3 DW	3 × 3 DW	1	6	32 × 32 × 96
	ConvNext	32 × 32 × 96	3 × 3 DW	–	1	4	32 × 32 × 96
	ConvNext	32 × 32 × 96	3 × 3 DW	–	1	4	32 × 32 × 96
Layer6	IB	32 × 32 × 96	–	3 × 3 DW	2	6	16 × 16 × 160
	IB	16 × 16 × 160	–	3 × 3 DW	1	8	16 × 16 × 160
	IB	16 × 16 × 160	–	3 × 3 DW	1	8	16 × 16 × 160
	IB	16 × 16 × 160	–	3 × 3 DW	1	8	16 × 16 × 320
Layer7	Conv	16 × 16 × 320	1 × 1 Conv	–	1	–	16 × 16 × 1280

2.2 Improved 3-SE Attention Mechanism

Seam defects are elongated and complex, which makes the fusion of multi-scale features prone to aliasing and unstable boundaries. To address this, we propose the improved 3-SE attention mechanism, an enhancement based on Squeeze-and-Excitation Networks (SENet), where the standard Squeeze and Excitation (SE) module structure, shown in Fig. 3a. This mechanism incorporates boundary-weighted statistics and branch-specific competitive normalization to selectively adapt multi-scale features, improving boundary detection and reducing feature dilution.

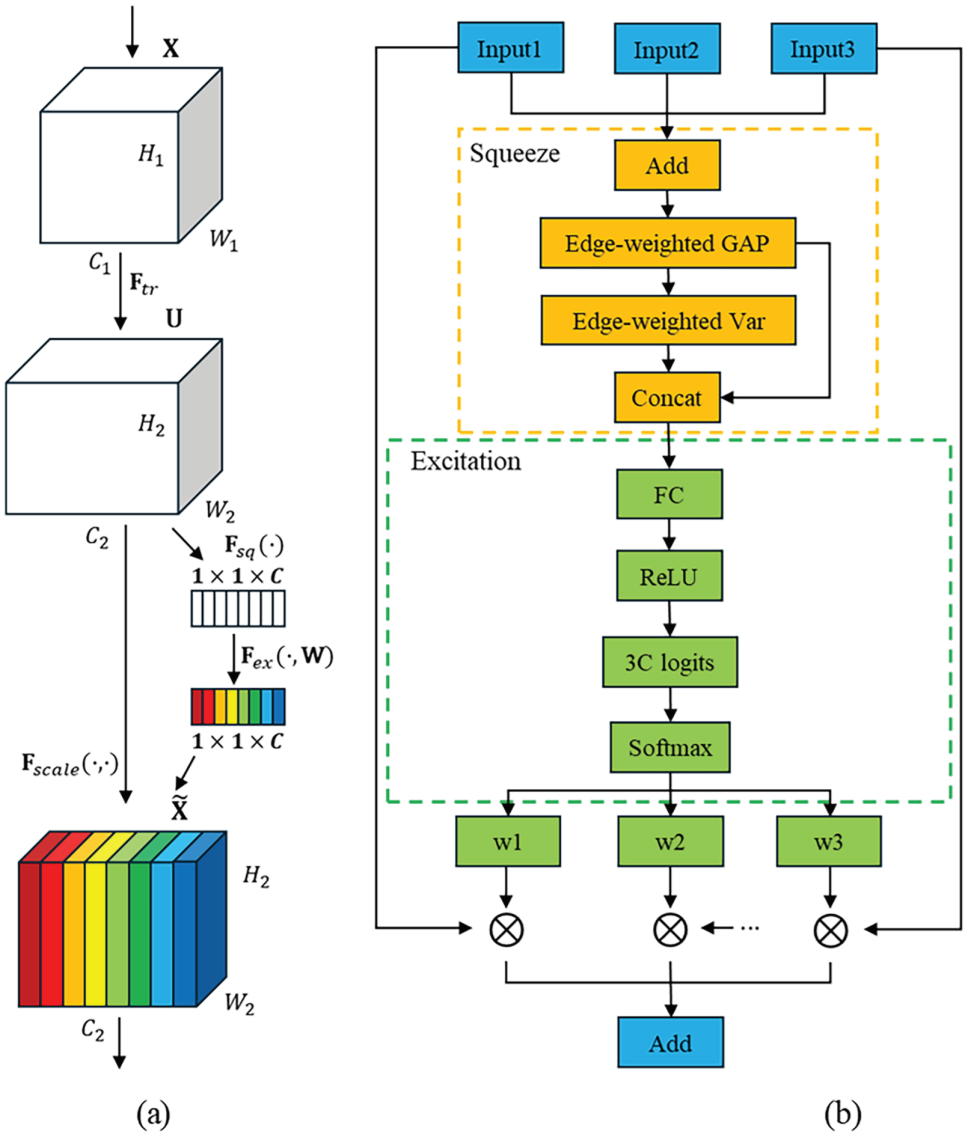

Figure 3: Standard SE module structure (**a**); Proposed 3-SE module structure (**b**)

Given three features to be fused, $F_1, F_2, F_3 \in \mathbb{R}^{B\times C\times H\times W}$, we first perform element-wise addition to obtain the fused feature:

$$F_{sum} = F_1 + F_2 + F_3 \tag{1}$$

Using the fused feature F_{sum}, a boundary weight map $E \in \mathbb{R}^{B\times C\times H\times W}$ is generated. This map extracts edge clues through shallow convolutions and nonlinear mapping, and obtains pixel-level weight distributions via Sigmoid normalization. The process assigns higher weights to seam-head edge pixels while setting background region weights close to zero. Subsequently, weighted mean and variance statistics are computed for each channel using this weight map:

$$\mu(c) = \frac{\sum_{h,w} E(h,w) F_{sum}(c,h,w)}{\sum_{h,w} E(h,w) + \varepsilon} \tag{2}$$

$$\mathrm{var}(c) = \frac{\sum_{h,w} E(h,w)\left(F_{sum}(c,h,w) - \mu(c)\right)^2}{\sum_{h,w} E(h,w) + \varepsilon} \tag{3}$$

where ε is the numerical stabilization term. The mean $\mu(c)$ describes the overall activation level of the channel in the boundary region, while the variance $\mathrm{var}(c)$ characterizes the dispersion of the activation distribution. These two components are concatenated to form the channel descriptor:

$$z = [\mu; \mathrm{var}] \in \mathbb{R}^{B\times 2C} \tag{4}$$

This explicitly enhances the contribution of the boundary region in the statistical features. Inputting the descriptor z into a two-layer fully connected network with Rectified Linear Unit (ReLU) activation yields a three-branch scoring vector:

$$[s_1, s_2, s_3] = \mathrm{FC}_2\left(\mathrm{ReLU}\left(\mathrm{FC}_1(z)\right)\right) \in \mathbb{R}^{B\times 3C} \tag{5}$$

This is reshaped to $[B, 3, C, 1, 1]$ and undergoes Softmax normalization along the branch dimension:

$$\omega_i(c) = \frac{\exp\left(s_i(c)\right)}{\sum_{k=1}^{3} \exp\left(s_k(c)\right)} \tag{6}$$

$$\sum_{i=1}^{3} \omega_i(c) = 1 \tag{7}$$

Obtain per-channel weights $\omega_1, \omega_2, \omega_3$ for the three feature streams. Finally, weight and sum each branch feature per channel:

$$\hat{F}_i = \omega_i \odot F_i \tag{8}$$

$$F_{out} = \hat{F}_1 + \hat{F}_2 + \hat{F}_3 \tag{9}$$

As shown in Fig. 3b, the Improved 3-SE attention mechanism reduces feature dilution during feature fusion by establishing competitive relationships. Each channel adaptively selects the most relevant scale features, enhancing the response and structural continuity in the seam-head boundary region, which may improve segmentation accuracy, particularly in low-contrast areas.

2.3 DASF Module

The width and orientation of seam defects vary, and if the receptive field does not match the target scale, boundary discontinuities and fine details may be lost. The original ASPP module uses a fixed dilation rate, making it ineffective for processing both fine and large-scale seam structures, while its multi-branch convolutions lead to high computational costs. To improve multi-scale adaptability for seam heads, We propose the DASF module. It introduces dynamic dilation rate adjustments and channel sliding-window strategies to model multi-scale contexts and enhance feature continuity.

As shown in Fig. 4, DASF extends the multi-branch parallelism of ASPP with three branches: the first uses a 1 × 1 convolution to compress and retain local information; the second applies channel-grouped sliding-window convolution with dynamic dilation to enhance multi-scale perception; and the third captures large-scale context through global average pooling. The three feature streams are weighted and fused using the 3-SE attention mechanism to generate feature maps with improved contextual continuity.

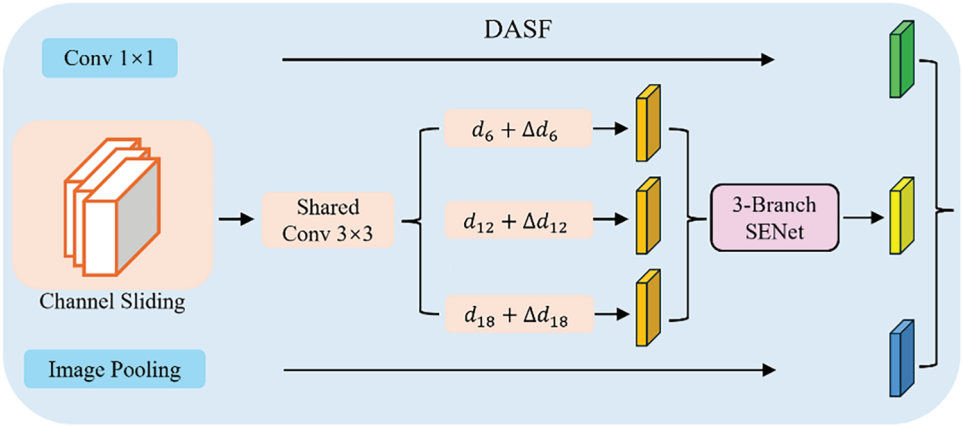

Figure 4: DASF module structure

As shown in Fig. 5, the second branch is the core of DASF. Unlike fixed dilated convolutions, it partitions feature maps using overlapping sliding windows, which reduces parameter overhead while improving feature coherence across channels. These overlapping windows help reduce semantic fragmentation and promote better coherence between local and global features.

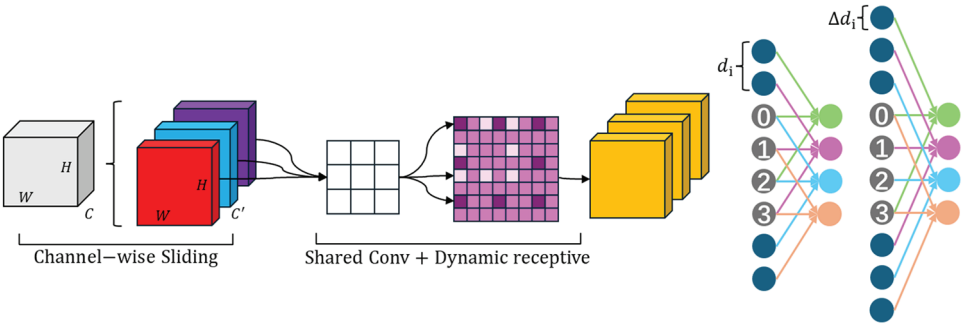

Figure 5: Second branch of the DASF module

For the high-level features $X \in \mathbb{R}^{B\times C\times H\times W}$ output by the backbone network, this module partitions them into multiple sub-channel blocks $\{X_i\}_{i=1}^{n}$. Each sub-block has dimensions $X \in \mathbb{R}^{B\times C'\times H\times W}$, where $C' < C$. The window size and stride are adjustable (e.g., 160 and 80) to accommodate different resolutions, with overlapping regions enhancing information exchange.

Additionally, a learnable dilation rate offset is introduced for each convolutional branch:

$$\hat{d}_i = d_i + \Delta d_i \tag{10}$$

here, d_i represents the initial dilation rate, and Δd_i denotes the learnable offset parameter. The initial dilation rates are set to $\{6, 12, 18\}$ for the three branches, respectively. The learnable offset parameters are initialized from a normal distribution with a mean of 0 and a standard deviation of 0.1, with the dilation rates constrained within ±1 of their initial values to ensure stability and computational efficiency. The design of the DASF module allows the receptive field to adapt to variations in seam width and orientation, alleviating issues of boundary discontinuities and detail loss that might occur with fixed dilation rates.

2.4 PLFF Module

Seam defect boundaries are slender and highly sensitive to positioning accuracy. Directly concatenating deep and shallow features can cause semantic conflicts and noise, resulting in blurred boundaries and inaccurate localization. The original DeepLabV3+ decoder relies on a single shallow feature layer for edge compensation, leading to insufficient multi-scale information integration. To address this, we propose the PLFF module, which uses a two-step fusion strategy to resolve semantic discrepancies, as illustrated in Fig. 6a. In the feature preparation stage, PLFF extracts three low-level feature maps from the UIB-MobileNetV2 backbone: primary (Layer1, 256 × 256), intermediate (Layer2, 128 × 128), and deep-low-level (Layer3, 128 × 128) features. These are compressed into 24 channels using Depthwise Separable Convolution (DPConv, Fig. 6b) and unified to 128 × 128 resolution via bilinear interpolation for consistency.

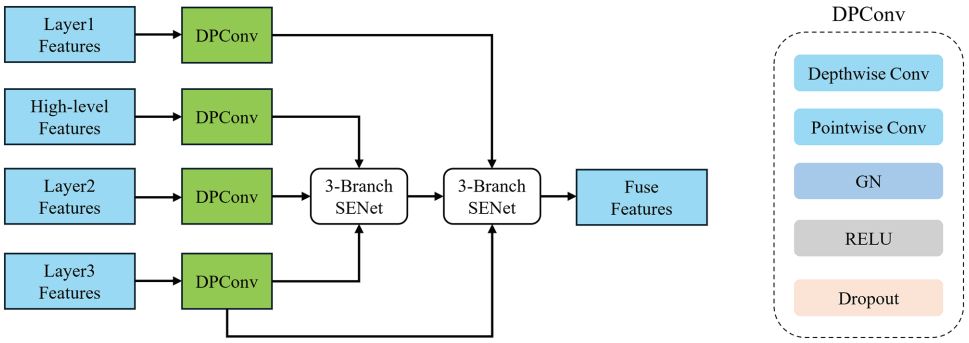

Figure 6: PLFF module structure (**a**); DPConv structure (**b**)

For progressive fusion, high-level features from the DASF module are compressed to 24 channels using DPConv and fused with the intermediate and deep-low-level features using the 3-SE attention mechanism. In the second stage, the refined features are further fused with the primary and intermediate low-level features to restore spatial details and enhance boundary localization. The intermediate feature is reused in both stages to ensure consistency. Finally, the fused shallow-enhanced features are concatenated with the high-level features and passed into the decoder for final prediction.

3 Experimental Environment and Evaluation Criteria

3.1 Experimental Environment

All experiments were conducted on Windows 10 with Python 3.8 and the PyTorch 2.4.1 framework, utilizing CUDA 12.4. The hardware configuration included an Intel Core i7-14700KF processor and an NVIDIA GeForce RTX 4070 Ti SUPER (16 GB VRAM).

The training process consisted of two stages. In the initial stage, the backbone parameters were frozen, and training focused solely on the top layers to facilitate rapid convergence. After 50 epochs, all layers were unfrozen, and the entire network was trained to further optimize performance. During the frozen phase, a smaller batch size was used, which was subsequently increased during the unfrozen phase.

All experiments were conducted using images with a resolution of 512 × 512 pixels, a batch size of 4, and a total of 200 training epochs. The Adam with Weight Decay (AdamW) optimizer, with a momentum of 0.9, was employed to help prevent overfitting. The learning rate started at 1×10^{-4} and decayed according to a cosine annealing schedule, with a minimum value of 1×10^{-6}. Random shuffling enabled to enhance robustness, and class weights were set to 1:2:3 to address class imbalance, ensuring balanced loss during training. Performance was evaluated on the validation set every 5 epochs, with early stopping applied if

validation performance did not improve over 5 consecutive evaluations. The model with the best validation performance was then selected for final testing.

3.2 Datasets

This study uses two datasets: the Fabric Roll Seam Defect (FSD) dataset from industrial production lines and the publicly available AITEX Fabric Defect Dataset. The FSD dataset contains 642 high-resolution fabric roll seam images, captured with a Hikvision MV-CA050-10GM camera controlled by a Beckhoff C6030 PLC. The dataset features various fabric backgrounds, with each image may containing 1 to 2 types of seams, oriented in different directions. Two types of seams are present: hem seams, which are typically narrower (1–2 mm), and joint seams, which are wider, reaching up to 15 mm, as shown in Fig. 7a. The AITEX dataset contains defects such as cracks and loose threads, divided into 256 × 256 sub-images for visualization and generalization evaluation, as shown in Fig. 7b. To improve model generalization and reduce overfitting, data augmentation techniques, including random rotations, blurring, salt-and-pepper noise, and brightness adjustment, were applied. The augmented dataset was split into training, validation, and test sets in an 8:1:1 ratio. All images in the FSD dataset were annotated using LabelMe, with annotations performed by the author to ensure consistency. Table 2 provides detailed information on the datasets.

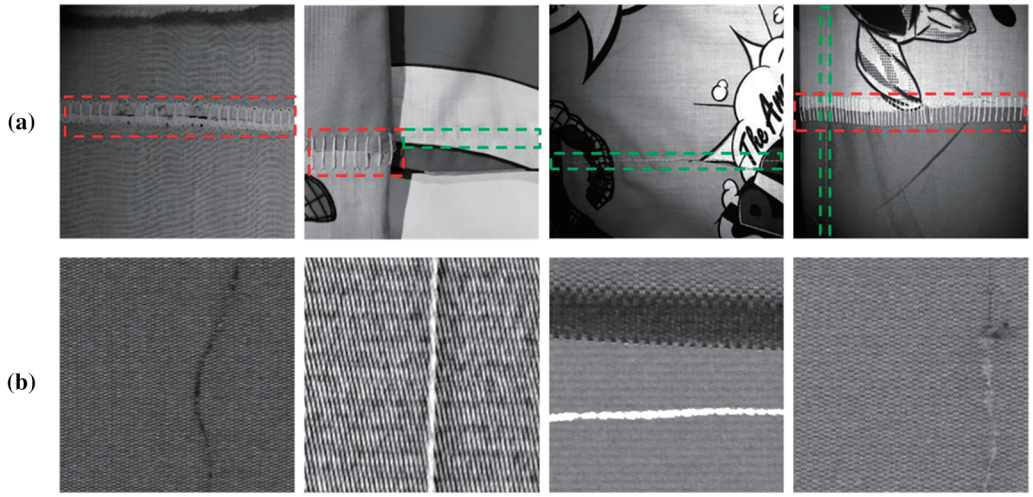

Figure 7: FSD dataset (**a**)—Red: Joint seams, Green: Hem seams; AITEX dataset (**b**)

Table 2: Detailed information on the datasets

Dataset	Image size	Training set	Validation set	Test set	Joint seam count	Hem seam count	Total
FSD	512 × 512	1576	176	176	1361	856	1928
AITEX	256 × 256	960	120	120	–	–	1200

3.3 Evaluation Criteria

To evaluate the model's segmentation performance, we use Intersection over Union (IoU), Mean Intersection over Union (mIoU), Mean Pixel Accuracy (mPA), and Boundary F-score (BF).

$$\mathrm{IoU} = \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}},\ \mathrm{mIoU} = \frac{1}{k+1}\sum_{i=0}^{k} \mathrm{IoU} \tag{11}$$

$$\mathrm{mPA} = \frac{1}{k+1}\sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}} \mathrm{TP}_b \tag{12}$$

$$\mathrm{P}_b = \frac{\mathrm{TP}_b}{\mathrm{TP}_b + \mathrm{FP}_b}, \mathrm{R}_b = \frac{\mathrm{TP}_b}{\mathrm{TP}_b + \mathrm{FN}_b} \tag{13}$$

$$\mathrm{BF} = \frac{2 \times \mathrm{P}_b \times \mathrm{R}_b}{\mathrm{P}_b + \mathrm{R}_b} \tag{14}$$

where, p_{ii} represents the number of pixels correctly classified as class i, p_{ij} represents the number of pixels from class incorrectly predicted as class j, $k+1$ represents the total number of classes, TP_b represents True Positives at the boundaries, FP_b represents False Positives at the boundaries, FN_b represents False Positives at the boundaries, P_b represents the proportion of true boundary pixels correctly predicted out of all predicted boundary pixels, R_b represents the proportion of true boundary pixels correctly predicted out of all true boundary pixels. IoU measures the overlap between predicted and actual regions, with higher values indicating better performance. mIoU is the average IoU across all categories, reflecting overall performance. mPA measures pixel-level accuracy. BF assesses boundary localization by combining precision and recall.

Frames per second (FPS) measures the number of images the network can process per second. FPS was tested with a batch size of 1, FP32 precision, and a 512 × 512 input. The test was run for 100 iterations, and the average inference time was calculated and converted to FPS.

4 Experimental Process and Analysis

This chapter presents experimental results to demonstrate the advantages of the proposed model in seam head segmentation.

4.1 Backbone Replacement Experiments

To evaluate the UIB-MobileNetV2 backbone for seam head segmentation, we compared it with baseline models: HRNet, ResNet50, ResNet101, MobileNetV2, and Xception using the FSD dataset. The fully connected layers and classification heads of each backbone were removed, leaving only the feature extraction component connected to the DeepLabV3+ decoder. As shown in Table 3, Xception achieved the highest accuracy (mIoU = 89.38%), but its large parameter count and computational overhead were prohibitive. Both MobileNetV2 and UIB-MobileNetV2 achieved accuracy close to Xception while maintaining a significantly smaller parameter size. Notably, UIB-MobileNetV2 achieved mIoU = 89.27% with 5.65 M parameters, and its per-class results were comparable to those of Xception, offering an optimal balance between segmentation accuracy and real-time performance.

4.2 DASF Module Comparison Experiments

To evaluate the DASF module's performance in seam head segmentation, we conducted comparative experiments on the FSD dataset using the UIB-MobileNetV2 backbone, analyzing both overall performance and internal design.

Table 3: Performance comparison of backbone networks

Backbone	Params Size/MB	GFLOPS/G	Params/M	mIoU/%	Joint IoU/%	Hem IoU/%	mPA/%
Hrnet	273.59	187.09	71.72	79.91	91.37	54.16	83.51
ResNet50	151.68	59.85	39.76	84.98	93.71	58.66	92.36
ResNet101	224.13	79.35	58.75	86.26	93.96	63.18	92.63
MobileNetV2	22.44	52.88	5.82	88.91	94.84	72.56	94.75
Xception	209.75	166.85	54.71	89.38	94.45	74.41	96.47
UIB-MobileNetV2	21.58	52.34	5.65	89.27	94.11	74.47	96.43

We conducted comparative experiments between the DASF and ASPP modules. As shown in Table 4, DASF reduces model parameters by 38.2%, decreases computational complexity by 8.6%, and improves mIoU from 89.27% to 90.92% compared to ASPP. Additionally, DASF reduces core branch parameters by 1.58M. These results show that DASF improves accuracy and efficiency while reducing model complexity.

Table 4: Comparison of DASF and ASPP modules

Baseline	Params size/MB	Params/M	GFLOPS/G	mIoU/%	Core branch params/M
ASPP	21.58	5.65	52.34	89.27	2.21
DASF	13.34	3.49	47.83	90.92	0.63

For internal design, we compared the performance of fixed and learnable configurations for the DASF module across different dilation rate combinations, as shown in Table 5. The learnable dynamic dilation rate consistently outperformed the fixed configuration, reducing edge discontinuities and detail loss, and improving the model's adaptability to seam heads of varying scales. The learnable setting of {6, 12, 18} achieved the best performance (mIoU = 90.92%) and was chosen as the default configuration for the DASF module.

Table 5: Impact of fixed vs. learnable dilation rates on DASF module performance

Dilation rate	Dynamic	mIoU/%	Joint IoU/%	Hem IoU/%
{6, 12, 18}	Fixed	89.78	93.80	76.37
{6, 12, 18}	Learnable	90.92	94.68	78.79
{4, 8, 12}	Fixed	90.03	93.21	77.78
{4, 8, 12}	Learnable	90.45	93.86	78.29
{3, 9, 15}	Fixed	89.68	93.87	75.99
{3, 9, 15}	Learnable	90.20	93.82	77.60

4.3 Ablation Studies

The ablation results show the impact of each module on MSC-DeepLabV3+'s performance. As shown in Table 6, baseline model, using MobileNetV2 as the backbone in DeepLabV3+, achieved 88.91% mIoU, indicating limited boundary perception. Adding the UIB-MobileNetV2 backbone improved mIoU to 89.27%, enhancing feature extraction. The DASF module further increased mIoU to 90.92%, improving structural continuity. The PLFF module contributed to boundary restoration, raising mIoU to 90.81%, though

it slightly reduced performance compared to DASF. Combining all modules led to the best performance, with mIoU of 92.30% and BF of 92.54%, demonstrating the complementary contributions of the modules. This combination achieved optimal performance while reducing parameters and GFLOPS. Additionally, we compared the performance of the model with and without data augmentation based on the improvements proposed in this paper. As shown in Table 7, the model with data augmentation outperforms the model without augmentation in both mIoU and BF metrics.

Table 6: Ablation experiments with different modules

Improved backbone	DASF	PLFF	BF/%	Joint IoU/%	Hem IoU/%	mIoU/%	mPA/%	Params/M	GFLOPS/G
–	–	–	88.85	94.84	72.56	88.91	94.75	5.82	52.88
√	–	–	89.02	94.11	74.47	89.27	96.43	5.65	52.34
√	√	–	90.41	94.68	78.79	90.92	96.91	3.49	47.69
√	–	√	91.57	94.92	78.16	90.81	96.52	5.60	50.57
√	√	√	92.54	95.41	82.07	92.30	97.47	3.44	46.14

Table 7: Ablation experiments with data augmentation

Model	Data augmentation	mIoU/%	BF/%
MSC-DeepLabV3+	√	86.45	84.64
	–	92.30	92.54

4.4 Comparison with Mainstream Semantic Segmentation Models

To evaluate MSC-DeepLabV3+'s performance in seam segmentation, we compared it with models such as HRNet, PSPNet, UNet, SegFormer, DeepLabV3+, and SFC_DeepLabv3+ [28] using the FSD dataset, all under the same experimental configuration. As shown in Table 8, high-precision models like PSPNet (ResNet50), UNet (ResNet50), and DeepLabV3+ (Xception) achieve mIoU scores between 84% and 90%, but their parameter counts exceed 40 million, leading to high computational overhead and inference speeds below 110 FPS. Lightweight models such as PSPNet (MobileNetV2) achieve, but with significantly reduced segmentation accuracy. SegFormer-B1 achieves mIoU = 90.47% and BF = 91.83%, demonstrating good accuracy, but its parameter size is still significant. The SFC_DeepLabv3+, an improved lightweight segmentation model, achieves FPS = 185 and BF = 91.59%, but still lags behind MSC-DeepLabV3+. In contrast, MSC-DeepLabV3+ achieves 92.30% mIoU and 92.54% BF at 273 FPS, outperforming existing methods in both accuracy and efficiency. To better visualize the comparison, we generated an mIoU–FPS performance distribution plot for both the "accuracy-first" and "speed-first" strategies. As shown in Fig. 8, MSC-DeepLabV3+ occupies the optimal region in both strategies, delivering both high accuracy and high speed.

Table 8: Comparison of model efficiency and performance

Model	Backbone	Params/M	Params size/MB	GFLOPS/G	FPS	mIoU/%	BF/%
HRNet	HRNetV2	9.64	37.63	32.81	67	88.39	85.85
PSPNet	ResNet50	46.71	178.17	118.43	166	84.18	81.93
PSPNet	MobileNetV2	2.38	9.06	6.031	429	81.78	79.99
UNet	VGG16	10.73	41.09	370.38	61	89.45	88.37
UNet	ResNet50	43.93	167.59	184.13	106	90.28	90.46
SegFormer	B1	13.68	52.18	26.495	167	90.47	91.83
DeepLabV3+	Xception	54.71	209.75	166.85	101	89.38	89.66
DeepLabV3+	MobileNetV2	5.82	22.44	52.88	309	88.91	88.85
SFC_DeepLabv3+	MobileNetV2	6.27	26.27	57.83	185	90.91	91.59
Ours	UIB-MobileNetV2	3.44	13.75	46.14	273	92.30	92.54

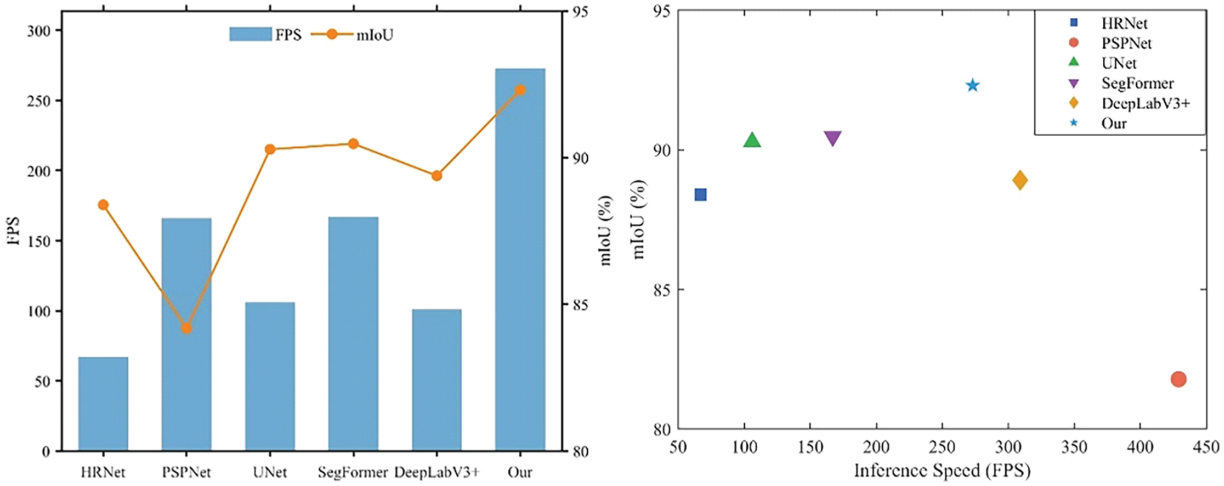

Figure 8: Comparison of "Accuracy-First" and "Speed-First" strategies

4.5 Visualization Comparison

To further validate the model's discriminative ability and reliability, we used the Grad-CAM visualization method to analyze feature activation responses. This technique generates heatmaps that highlight the regions of interest identified during deep neural network inference. As shown in the Fig. 9, the regions captured by MSC-DeepLabV3+ closely align with the actual distribution of seam heads, effectively emphasizing slender boundaries. This confirms that the model's predictions are based on target-relevant features rather than background noise. These results not only validate the model's reliability but also enhance its interpretability, providing valuable insights for future optimization and industrial deployment.

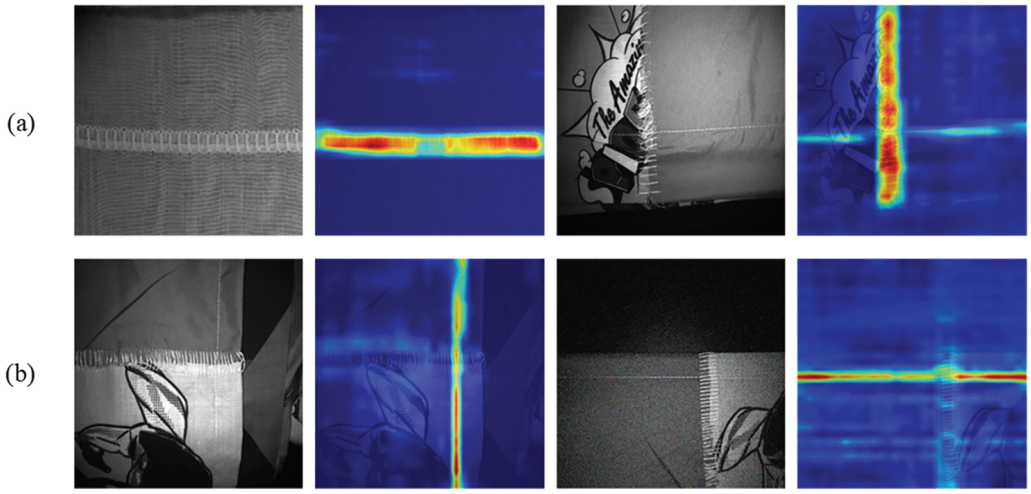

Figure 9: Overlapped heatmap (**a**) Joint seams heatmaps (**b**) Hem seams heatmaps

To visually compare segmentation performance across various seam-head scenarios and elongated defects, we present visualization results from models trained on the FSD and AITEX datasets and tested on their respective datasets. Fig. 10 shows that existing mainstream methods often suffer from edge discontinuities and local missed detections in weakly textured or slender regions, resulting in fragmented or inconsistent contours. In contrast, the proposed MSC-DeepLabV3+ generates segmentation maps with continuous boundaries and well-defined structures, effectively suppressing background interference and false positives. Moreover, it shows stable performance on the AITEX dataset, indicating a promising generalization ability across diverse scenarios.

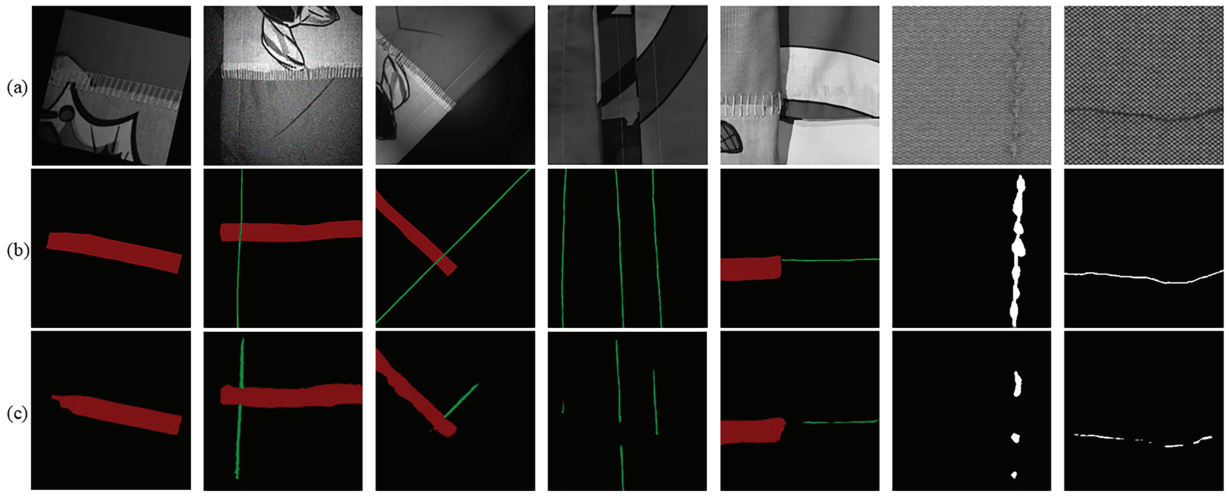

Figure 10: (Continued)

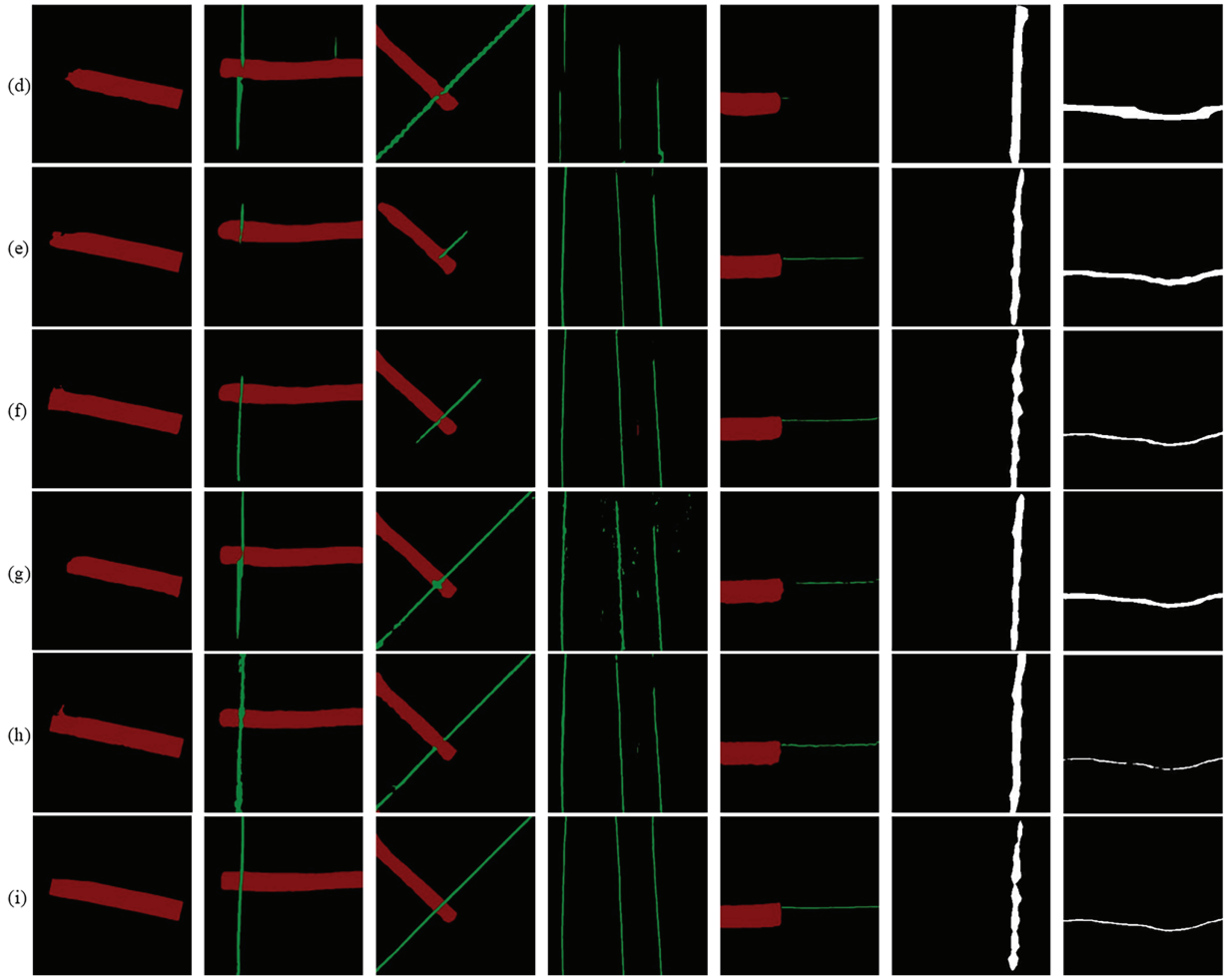

Figure 10: Comparison of different methods for visualization (**a**) Original (**b**) Label (**c**) Hrnet (**d**) Pspnet (**e**) Unet (**f**) Segformer (**g**) DeepLabV3+ (Xception) (**h**) DeepLabV3+ (MobileNetV2) (**i**) Ours

4.6 Feasibility Analysis for Model Deployment

Conducting a feasibility analysis during the design of deep learning models is crucial for assessing their adaptability and effectiveness in practical applications. Smaller model sizes and lower computational costs lead to faster inference speeds, making models more suitable for deployment on resource-constrained hardware. The model proposed in this paper, with a 512 × 512 input resolution, uses approximately 815.83 MB of memory, enabling stable operation on embedded devices, as shown in Table 9.

We evaluated the model's inference speed across multiple hardware platforms, with results shown in in Table 10. The model achieves 43 FPS even on the Jetson Orin Nano platform. The number of parameters affects both computational cost and training time, while the model size reflects its memory footprint. Smaller models with fewer parameters generally offer faster inference, making them suitable for real-time applications. In industrial settings, detecting a single image in under 33 ms (i.e., over 30 FPS) qualifies as real-time detection. Based on these results, our model is theoretically capable of meeting real-time detection requirements.

Table 9: Resource consumption for model deployment

Input size (MB):	**3.15**
Forward/backward pass size (MB):	798.90
Params size (MB):	13.79
Estimated total size (MB):	815.83
GFLOPS(G):	46.14
Params (M):	3.44

Table 10: Comparison of FPS for different embedded platforms

Platform	**Jetson orin NX**	**Google coral TPU**	**Jetson orin nano**
FPS	130	52	43

5 Conclusion

In this paper, we introduce MSC-DeepLabV3+, based on DeepLabV3+, with UIB-MobileNetV2 as the backbone network and integrated DASF and PLFF modules. The model addresses several challenges in fabric roll seam detection while significantly reducing computational complexity. Through comprehensive ablation experiments, we demonstrate the individual and combined contributions of these modules to segmentation performance. After integrating all enhancements, the model achieves relative improvements of 3.04% in mIoU and 3.14% in BF. Comparisons with mainstream segmentation models on our dataset show that MSC-DeepLabV3+ achieves mIoU and BF of 92.30% and 92.54%, respectively, outperforming these models. Furthermore, MSC-DeepLabV3+ also exhibits advantages in parameter efficiency, model size, and inference speed.

However, there are still some limitations. Firstly, the DASF module, while enhancing multi-scale adaptability with dynamic dilation rates, may fail when seam structures deviate from the training data or when background patterns resemble finer fabric roll seams. Additionally, during feature fusion, the model tends to focus on prominent seams, potentially overlooking subtle details, which could affect detection accuracy in complex production scenarios. The current implementation has not been fully evaluated under varying conditions such as lighting changes, occlusion, or different fabric textures, and the model's robustness in these scenarios requires further validation. Moreover, the FSD dataset used in this study is relatively limited, and future work should expand it to include a broader range of fabrics and seam types. Fabric anomalies, such as broken threads and loose ends, may also interfere with detection and impact model accuracy.

Future work will involve pre-training the model on a wider variety of fabric and seam types to enhance its segmentation and detection capabilities in real-world applications. We will also explore open-set defect detection, enabling the model to handle defect types not present in the training data, thus improving its generalization ability. Additionally, we plan to optimize the model through techniques such as quantization and pruning, making it more suitable for deployment on industrial devices and further enhancing its applicability in real-time industrial monitoring. Furthermore, we are committed to integrating motion control and machine vision on the same platform, which will achieve excellent real-time performance, significantly improve device efficiency, and minimize redundant delays in motion and robotic control.

Acknowledgement: The authors gratefully acknowledge the institutional support provided by Zhejiang Sci-Tech University for this research. Special thanks are extended to colleagues who provided valuable technical assistance and constructive feedback.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Kuntao Lv, Weimin Shi; model development and implementation: Kuntao Lv; data collection and annotation: Kuntao Lv, Chang Xuan; analysis and interpretation of results: Kuntao Lv, Chang Xuan, Ji Wu; draft manuscript preparation: Kuntao Lv; critical revision and editing: Weimin Shi, Ji Wu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Due to commercial considerations, these data are not publicly available, but can be obtained from the corresponding author upon reasonable request. The third-party dataset used in this study can be accessed through the AITEX FABRIC IMAGE DATABASE—Aitex website.

Ethics Approval: This research did not involve human participants or animals; therefore, ethical approval was not required.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Asha V, Bhajantri NU, Nagabhushan P. Automatic detection of texture defects using texture-periodicity and Gabor wavelets. In: Computer networks and intelligent computing. Berlin/Heidelberg, Germany: Springer; 2011. p. 548–53. doi:10.1007/978-3-642-22786-8_69.
2. Ngan HYT, Pang GKH, Yung NHC. Automated fabric defect detection—a review. Image Vis Comput. 2011;29(7):442–58. doi:10.1016/j.imavis.2011.02.002.
3. Wu Y, Zhou J, Akankwasa NT, Wang K, Wang J. Fabric texture representation using the stable learned discrete cosine transform dictionary. Text Res J. 2019;89(3):294–310. doi:10.1177/0040517517743688.
4. Wang J, Wang W, Zhang Z, Lin X, Zhao J, Chen M, et al. YOLO-DD: improved YOLOv5 for defect detection. Comput Mater Continua. 2024;78(1):759–80. doi:10.32604/cmc.2023.041600.
5. Khwakhali US, Tra NT, Tin HV, Khai TD, Tin CQ, Hoe LI. Fabric defect detection using gray level co-occurence matrix and local binary pattern. In: Proceedings of the 2022 RIVF International Conference on Computing and Communication Technologies (RIVF); 2022 Dec 20–22; Ho Chi Minh City, Vietnam. p. 226–31. doi:10.1109/RIVF55975.2022.10013920.
6. Pourkaramdel Z, Fekri-Ershad S, Nanni L. Fabric defect detection based on completed local quartet patterns and majority decision algorithm. Expert Syst Appl. 2022;198(5):116827. doi:10.1016/j.eswa.2022.116827.
7. Chang X, Liu W, Zhu C, Zou X, Gui G. Bilayer Markov random field method for detecting defects in patterned fabric. J Circuits Syst Comput. 2022;31(3):2250058. doi:10.1142/s021812662250058x.
8. Brad R, Modrângă C, Brad R. Fabric defect detection using Fourier transform and Gabor filters. J Text Eng Fash Technol. 2017;3(4):684–8. doi:10.15406/jteft.2017.03.00107.
9. Barman J, Wu HC, Kuo CJ. Development of a real-time home textile fabric defect inspection machine system for the textile industry. Text Res J. 2022;92(23–24):4778–88. doi:10.1177/00405175221111477.
10. Sadaghiyanfam S. Using gray-level-co-occurrence matrix and wavelet transform for textural fabric defect detection: a comparison study. In: Proceedings of the 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT); 2018 Apr 18–19; Istanbul, Turkey. p. 1–5. doi:10.1109/EBBT.2018.8391440.
11. Yıldız K, Buldu A, Demetgul M. A thermal-based defect classification method in textile fabrics with K-nearest neighbor algorithm. J Ind Text. 2016;45(5):780–95. doi:10.1177/1528083714555777.
12. Jing J, Li H, Li P. Combined fabric defects detection approach and quadtree decomposition. J Ind Text. 2012;41(4):331–44. doi:10.1177/1528083711419877.

13. Xu Z, Bao Y, Tian B. Improved YOLOv8 garment sewing defect detection method based on attention mechanism. J Meas Eng. 2024;12(4):706–21. doi:10.21595/jme.2024.24283.
14. Guo Y, Kang X, Li J, Yang Y. Automatic fabric defect detection method using AC-YOLOv5. Electronics. 2023;12(13):2950. doi:10.3390/electronics12132950.
15. Yang Z, Liu S, Hu H, Wang L, Lin S. RepPoints: point set representation for object detection. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 9656–65. doi:10.1109/iccv.2019.00975.
16. Li P, Xu F, Wang J, Guo H, Liu M, Du Z. DGConv: a novel convolutional neural network approach for weld seam depth image detection. Comput Mater Contin. 2024;78(2):1755–71. doi:10.32604/cmc.2023.047057.
17. Hao S, Zhou Y, Guo Y. A brief survey on semantic segmentation with deep learning. Neurocomputing. 2020;406(9):302–21. doi:10.1016/j.neucom.2019.11.118.
18. Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell. 2017;39(4):640–51. doi:10.1109/TPAMI.2016.2572683.
19. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention—MICCAI 2015. Cham, Switzerland: Springer; 2015. p. 234–41. doi:10.1007/978-3-319-24574-4_28.
20. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 July 21–26; Honolulu, HI, USA. p. 6230–9. doi:10.1109/CVPR.2017.660.
21. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer vision—ECCV 2018. Cham, Switzerland: Springer; 2018. p. 833–51. doi:10.1007/978-3-030-01234-2_49.
22. Guo Y, Xiao Z, Geng L. Defect detection of 3D braided composites based on semantic segmentation. J Text Inst. 2023;114(4):574–83. doi:10.1080/00405000.2022.2054103.
23. Hu S, Zhang J. Modeling of fabric sewing break detection based on U-Net network. Text Res J. 2024;94(23–24):2695–706. doi:10.1177/00405175241259204.
24. Kamann C, Rother C. Benchmarking the robustness of semantic segmentation models. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 8825–35. doi:10.1109/cvpr42600.2020.00885.
25. Nguyen QT. Defective sewing stitch semantic segmentation using DeeplabV3+ and EfficientNet. Intel Artif. 2022;25(70):64–76. doi:10.4114/intartif.vol25iss70pp64-76.
26. Bai Z, Jing J. Mobile-Deeplab: a lightweight pixel segmentation-based method for fabric defect detection. J Intell Manuf. 2024;35(7):3315–30. doi:10.1007/s10845-023-02205-1.
27. Liu Y, Shen J, Ye R, Wang S, Ren J, Pan H. FP-Deeplab: a segmentation model for fabric defect detection. Meas Sci Technol. 2024;35(10):106008. doi:10.1088/1361-6501/ad5f50.
28. Xia Y, Qiu J. SFC_DeepLabv3+: a lightweight grape image segmentation method based on content-guided attention fusion. Comput Mater Contin. 2025;84(2):2531–47. doi:10.32604/cmc.2025.064635.