Textile & Leather Review

ISSN 2623-6281| www.tlr-journal.com | 10.31881/TLR

Deep-Learning-Based Probabilistic Graphical Models for Automated Defect Detection in Smart Textile Manufacturing

Guang Gao, Chuangchuang Chen

How to cite: Gao G, Chen C. Deep-Learning-Based Probabilistic Graphical Models for Automated Defect Detection in Smart Textile Manufacturing. Textile & Leather Review. 2025; 8:925-939. https://doi.org/10.31881/TLR.2025.925

How to link: https://doi.org/10.31881/TLR.2025.925

Published: 10 December 2025

Deep-Learning-Based Probabilistic Graphical Models for Automated Defect Detection in Smart Textile Manufacturing

Guang GAO*, Chuangchuang CHEN
School of Artificial Intelligence, Zhoukou Normal University, Zhoukou 466000, Henan, China
*ggao198@163.com

Article
https://doi.org/10.31881/TLR.2025.925
Received 8 August 2025; Accepted 28 August 2025; Published 10 December 2025

ABSTRACT
Smart textile manufacturing is challenged by unique and subtle defects that are difficult to detect using traditional inspection methods. To address these challenges, we propose a novel hybrid framework, Deep-CRF, that integrates the feature-extraction capabilities of deep learning (DL) with the contextual reasoning of probabilistic graphical models (PGMs) for high-accuracy automated defect detection. Our approach uses a ResNet-50 model, fine-tuned on a custom Smart Textile Defect Dataset (STDD), to produce an initial defect probability map. This map is then processed by a fully-connected conditional random field (CRF), which refines the segmentation mask by modeling spatial dependencies — thereby improving boundary accuracy and reducing prediction noise. Experimental results demonstrate the superiority of the Deep-CRF framework over traditional methods and standard DL models such as U-Net. Our model achieved a state-of-the-art mean intersection over union (mIoU) of 93.7% and an F1 score of 94.1%. The CRF refinement stage proved crucial, improving the mIoU by 3.5 percentage points over the ResNet-50 baseline. This work presents a robust and accurate solution that can significantly enhance quality control and pave the way for reliable, large-scale production of smart textiles.

KEYWORDS
defect detection, deep learning (DL), convolutional neural networks (CNNs), probabilistic graphical models (PGMs), conditional random field (CRF)

INTRODUCTION

The convergence of electronics and textiles has given rise to smart textiles, also known as e-textiles, which are fabrics embedded with electronic components [1]. These materials can sense, react, and adapt to environmental stimuli, enabling applications in healthcare monitoring (e.g., ECG- and EMG-sensing textiles), protective clothing (e.g., temperature-regulating suits), and human–computer interaction [2–4]. The manufacturing of smart textiles involves complex processes such as integrating conductive yarns, embedding fiber optics, and printing functional electronic materials onto fabric substrates [5].

Despite their immense potential, the large-scale industrialization of smart textiles is hindered by significant challenges in quality control. The heterogeneity of materials (e.g., metallic yarns, polymers, natural fibers) and the complexity of integrated structures make the manufacturing process susceptible to a unique set of defects not encountered in conventional textile production. These defects can be categorized as structural (e.g., conductive yarn breakage, short circuits between conductive paths) and functional (e.g., electrode delamination, non-uniform coating of functional materials), as shown in Figure 1a–d. Such defects, even if microscopic, can compromise the functionality of the final product, leading to significant economic losses and potential safety risks [6].

Traditional defect detection in the textile industry relies heavily on manual inspection, which is subjective, fatiguing, and unsuitable for the high-throughput demands of modern manufacturing [7]. While conventional machine-vision systems using techniques such as Gabor filters, Fourier analysis, or statistical methods have been applied to regular textiles, they struggle with the complexity of smart textiles [8]. These methods often fail to adapt to the high variability in texture, color, and defect morphology present in e-textiles.

In recent years, deep learning (DL), particularly convolutional neural networks (CNNs), has achieved state-of-the-art performance in various computer vision tasks, including industrial defect detection [9]. CNNs can automatically learn hierarchical feature representations from raw pixel data, making them highly effective for identifying complex patterns. Architectures such as U-Net [10] and Faster R-CNN [11] have been successfully applied to detect defects in steel, semiconductors, and conventional fabrics.

However, applying these models directly to smart textiles presents two primary challenges. (1) Subtlety and ambiguity. Defects such as minor delamination or slight non-uniformity in a conductive coating can be visually subtle and difficult to distinguish from natural texture variations, leading to high false-positive or false-negative rates for standard CNNs. (2) Lack of contextual awareness. Standard CNNs typically perform pixel-wise or region-wise classification independently and often fail to leverage spatial relationships between pixels, resulting in noisy and fragmented segmentation masks. For instance, a broken conductive yarn is a continuous line defect, but a CNN might detect only parts of it, yielding an incomplete representation.

Consequently, these unique challenges imply that existing public datasets for conventional textile inspection, which typically feature defects such as slubs, holes, or color stains on homogeneous fabric patterns, are fundamentally inadequate. They lack specific and complex defect classes—such as functional material delamination or micro-cracks in integrated fibers—that are critical for quality control in smart textile manufacturing.

To address these limitations, this paper proposes a hybrid approach that integrates the feature-learning ability of CNNs with the contextual reasoning capabilities of probabilistic graphical models (PGMs). Specifically, we employ a conditional random field (CRF) as a post-processing module for the output of a deep CNN. PGMs excel at modeling dependencies between variables [12]; in this context, a CRF can enforce spatial smoothness in the defect probability map, encouraging adjacent pixels to share the same label unless there

is strong evidence to the contrary. This allows the model to refine defect boundaries, eliminate isolated prediction errors, and produce structurally consistent segmentation masks.

The main contributions of this study are as follows. First, we propose a novel Deep-CRF framework tailored for automated defect detection in smart textiles, combining a ResNet-50 backbone for feature extraction with a fully-connected CRF for contextual refinement. Second, we construct and introduce a specialized Smart Textile Defect Dataset (STDD) comprising high-resolution images of various smart textiles with meticulously annotated defects to serve as a benchmark for this research area. Third, we conduct extensive experiments demonstrating that our proposed framework significantly outperforms traditional methods and standard deep learning models in terms of accuracy, precision, and robustness.

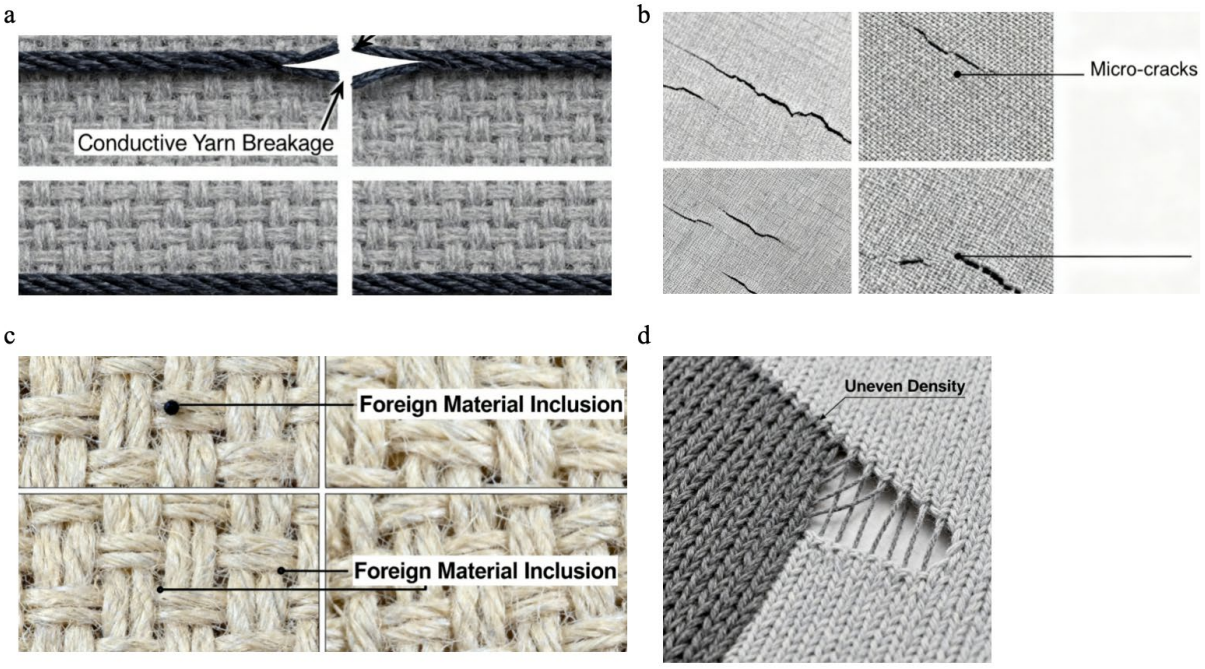

Figure 1. Schematic illustration of common defect types observed in smart textile manufacturing, highlighting their typical visual characteristics. (a) Conductive Yarn Breakage: Discontinuity in the integrated conductive pathway, leading to electrical open circuits. (b) Micro-cracks: Fine, irregular cracks appearing on the textile substrate, potentially impairing sensor functionality or mechanical integrity. (c) Foreign Material Inclusion: Unintended foreign particles or fibers embedded within the textile structure, which can cause short circuits or local stress concentrations. (d) Uneven Weave/Knitting Density: Inconsistent spacing or alignment of yarns, resulting in variations in mechanical properties, breathability, or electrical uniformity

RELATED WORK

Traditional Machine Vision for Textile Defect Detection

Before the advent of deep learning, research on automated textile defect detection was dominated by traditional machine-vision techniques. These approaches can be broadly classified into three categories: statistical, spectral, and model-based methods.

Statistical approaches analyze the texture properties of fabrics. Methods based on co-occurrence matrices [13] and local binary patterns (LBP) [14] extract features that describe the spatial arrangement of gray levels. These features are then fed into a classifier, such as a support vector machine (SVM), to distinguish between defective and non-defective regions. While effective for simple, periodic textures, these methods are sensitive to variations in lighting and fabric rotation.

Spectral approaches operate in the frequency domain. The Fourier transform is used to identify disruptions in the periodic pattern of the fabric, which often manifest as high-frequency components [15]. Gabor filters, a set of band-pass filters, have been widely used due to their ability to capture texture information at different scales and orientations [8]. However, these methods are less effective for non-periodic textures and struggle to localize defects accurately.

Model-based approaches attempt to create a mathematical model of the defect-free texture and identify deviations from this model. Markov random fields (MRFs) have been used to model local pixel dependencies in texture [16]. Autoregressive models have also been explored. These methods can be robust but are often computationally expensive and tailored to specific texture types. For smart textiles, with their inherent heterogeneity, building a single comprehensive texture model is extremely challenging.

Deep Learning for Defect Detection

Deep learning has revolutionized industrial inspection. CNN-based methods can be categorized into classification, detection, and segmentation approaches. Classification models [17] determine whether an entire image contains a defect but do not localize it. Object detection models, such as YOLO [18] and Faster R-CNN [11], place bounding boxes around defects. While useful, bounding boxes are often too coarse for precise analysis of defect shape and size.

For fine-grained analysis, semantic segmentation models are preferred. These models assign a class label (e.g., "defect" or "background") to every pixel in an image. The U-Net architecture [10], a popular CNN model known for its U-shaped encoder–decoder structure and skip connections, has become a de facto standard for biomedical and industrial image segmentation. Fully convolutional networks (FCNs) [19] are another popular choice. These models have been successfully applied to detect defects on surfaces like steel [20], solar cells, and conventional fabrics [21]. However, their pixel-wise predictions can be noisy and lack spatial consistency.

Hybrid DL–PGM Models

To overcome the limitations of pure CNN-based segmentation, researchers have explored combining CNNs with probabilistic graphical models (PGMs). The most common approach is the CNN–CRF model. Chen et al. [22] first proposed using a fully-connected CRF as a post-processing step to refine the coarse output of a CNN for semantic segmentation. The CNN provides a powerful unary potential (initial pixel-wise class probability), while the CRF models pairwise potentials, encouraging label consistency among spatially or color-wise similar pixels. This synergy has proven effective in various domains, including medical imaging and scene parsing.

In the context of industrial inspection, this hybrid approach is particularly promising. The structured nature of defects (e.g., cracks, scratches, breakages) lends itself well to contextual modeling by a CRF. While some works have applied this approach to general surface inspection, its application to the specific and complex domain of smart textiles remains largely unexplored. Our work aims to fill this gap by designing and validating a Deep-CRF framework specifically for the challenges posed by smart textile manufacturing.

PROPOSED METHODOLOGY

Framework Overview

The proposed defect-detection framework, Deep-CRF, is illustrated in Figure 2. The process consists of four main stages: (1) image acquisition and preprocessing, (2) deep feature extraction using a ResNet-50 backbone, (3) initial defect-segmentation mask generation, and (4) segmentation refinement using a fully-connected CRF.

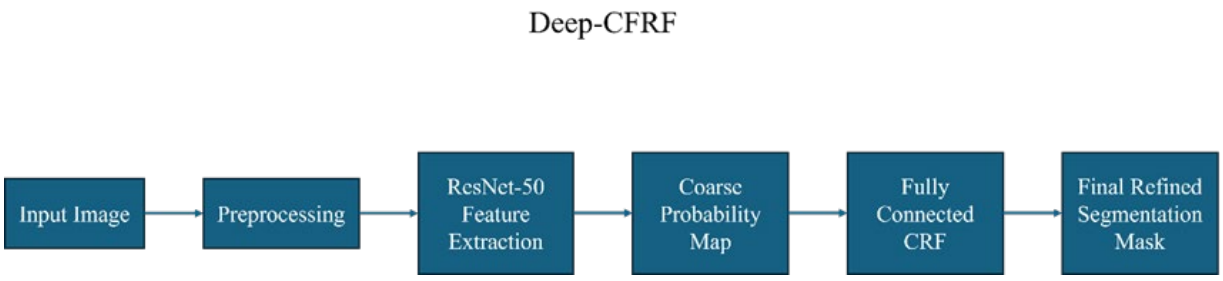

Figure 2. Architecture of the proposed Deep-CRF framework for smart textile defect detection

The input to the system is a high-resolution image of a smart textile sample. After preprocessing, the image is fed into the deep convolutional neural network (CNN), which outputs a coarse probability map indicating the likelihood of each pixel belonging to the defect class. This map serves as the unary term for the CRF. The CRF then optimizes a global energy function that incorporates both the CNN's predictions and a pairwise term that encourages spatial smoothness, resulting in a refined, accurate, and spatially coherent final defect mask.

Data Acquisition and Dataset (STDD)

A key contribution of this work is the creation of the Smart Textile Defect Dataset (STDD). To the best of our knowledge, there are no publicly available, annotated datasets specifically designed for the unique defect detection challenges in smart textiles. While datasets for conventional textile defects exist, they do not capture the unique failure modes that arise from the integration of electronics and fabrics, such as conductive yarn breakage, electrode delamination, or functional coating non-uniformity. These specific and subtle defects necessitate the creation of a specialized benchmark. While several public datasets for defect detection in conventional textiles exist, they are not suitable for the distinct challenges posed by smart textiles, rendering them inadequate for training robust models for this application. Existing datasets primarily focus on structural faults in homogenous, periodic textures (e.g., woven cotton or denim). In contrast, smart textiles are heterogeneous composites, integrating materials like metallic yarns, polymers, and functional coatings. This results in unique failure modes not found in traditional fabrics, such as conductive yarn breakage, electrode delamination, and non-uniform coating of functional materials. These defects are often characterized by low visual contrast and subtle textural variations that are easily confused with the fabric's intrinsic complex structure. We constructed it using a high-resolution line-scan camera system (2400 dpi) under controlled, diffuse LED illumination.

- Textile Samples: The dataset includes three representative types of smart textiles:
  - Conductive Yarn-Integrated Fabric: Woven polyester fabric with integrated silver-coated conductive yarns for signal transmission.
  - Fiber Optic Embedded Fabric: Jacquard fabric with embedded polymer optical fibers for strain sensing.
  - Functional Coated Fabric: Non-woven fabric with a screen-printed carbon-based coating for pressure sensing.
- Defect Categories: We cataloged and annotated five common and critical defect types:
  - D1: Conductive Yarn Breakage: A complete or partial break in the conductive yarn.
  - D2: Electrode Delamination: Peeling of the printed electrode from the fabric substrate.
  - D3: Uneven Coating: Non-uniform thickness or coverage of the functional material.
  - D4: Short Circuit: Unwanted contact between two adjacent conductive paths.
  - D5: Fiber Crack: Micro-cracks in the embedded optical fiber.

The STDD consists of 1,500 high-resolution images (4096 × 2048 pixels). We partitioned the original images into smaller patches of 512 × 512 pixels for training. This patch size was deliberately chosen to be larger than the network input size (256 × 256) to ensure each patch captures the entirety of a defect and its immediate surrounding context; directly cropping to 256 × 256 from the source images could fragment a single

meaningful defect across multiple patches and lose critical contextual information. After patch extraction, all 512 × 512 patches were resized to 256 × 256 pixels to meet the network input requirements.

To train and evaluate the model, the 1,500 original images were randomly partitioned at the image level into a training set (1,050 images, 70%), a validation set (150 images, 10%), and a test set (300 images, 20%). This partitioning, performed prior to patch extraction, prevents data leakage and ensures a fair evaluation of the model's generalization. All patches for training and validation were extracted exclusively from the corresponding images, and all test patches were generated only from the test images, guaranteeing no overlap between the sets.

Image Preprocessing

All image patches were resized to 256 × 256 pixels to meet the input requirements of the neural network. Pixel values were normalized to the range [0, 1]. For the subsequent CRF refinement, however, we operate on the original 8-bit RGB image (intensity range [0, 255]) and pixel coordinates in the 256 × 256 image grid, following Krähenbühl and Koltun [23]. The normalization step is only applied to the input of the CNN branch. To mitigate overfitting and increase model robustness, we applied data augmentation to the training set, including random horizontal and vertical flips, rotations from −15° to +15°, and slight brightness adjustments.

Deep Feature Extraction with ResNet-50

We adopt ResNet-50 [24] as the backbone of our segmentation network. ResNet-50 is a deep residual architecture that has proven highly effective at learning rich visual representations while alleviating the vanishing-gradient problem. To leverage prior knowledge from large-scale natural images, we initialize the backbone with weights pre-trained on the ImageNet dataset and fine-tune all layers on the STDD.

The original global average pooling and fully connected classification head are removed. The remaining convolutional blocks act as an encoder that produces a compact, high-level feature map from the 256 × 256 input patch. On top of this encoder we build a lightweight decoder that progressively upsamples the feature maps back to the original input resolution and outputs dense, pixel-wise predictions.

Each decoder block consists of an upsampling operation (implemented with a transposed convolution), followed by a 3 × 3 convolution and a ReLU non-linearity. Successive blocks gradually increase the spatial resolution while decreasing the channel dimensionality. A final 1 × 1 convolution maps the last decoder feature map to two channels (defect vs. background), and a softmax function produces the per-pixel class probabilities. The overall decoder configuration, including kernel sizes, strides and channel dimensions, is summarized in Table 1.

The network is trained end-to-end using a standard pixel-wise cross-entropy loss between the predicted probability map and the ground-truth segmentation mask. Let x_i denote the feature vector of pixel i at the

output layer, and $P(y_i|x_i)$ the predicted probability that pixel i belongs to class $y_i \in \{0,1\}$ (0 for background, 1 for defect). The output probability map P serves as the unary term for the subsequent CRF refinement stage.

Table 1. Architecture of the decoder path for the modified ResNet-50

Layer name	Operation	Kernel / stride	Output channels	Output size
Decoder Block 1	Transposed Conv + 3 × 3 Conv + ReLU	kernel 2 × 2, stride 2	1024	16 × 16
Decoder Block 2	Transposed Conv + 3 × 3 Conv + ReLU	kernel 2 × 2, stride 2	512	32 × 32
Decoder Block 3	Transposed Conv + 3 × 3 Conv + ReLU	kernel 2 × 2, stride 2	256	64 × 64
Decoder Block 4	Transposed Conv + 3 × 3 Conv + ReLU	kernel 2 × 2, stride 2	128	128 × 128
Decoder Block 5	Transposed Conv + 3 × 3 Conv + ReLU	kernel 2 × 2, stride 2	64	256 × 256
Output Layer	1 × 1 Convolution + Softmax	kernel 1 × 1, stride 1	2	256 × 256

Segmentation Refinement with Conditional Random Field (CRF)

Although the CNN produces reasonably accurate segmentation masks, its predictions can contain noise, small spurious regions and slightly misaligned boundaries, especially around fine-scale defects. To further improve spatial consistency, we refine the CNN output using a fully connected CRF.

Let X be the input image and $Y=\{y_1,y_2,\cdots,y_N\}$ be the set of label assignments for all N pixels, where $y_i \in \{0,1\}$. The CRF defines the Gibbs energy of a labeling Y conditioned on X as:

$$E(y/X)=\sum_i \psi_u\left(y_i/X+\sum_{i<j}\psi_p\left(y_i,y_j/X\right)\right) \quad (1)$$

where ψ_u is the unary potential and ψ_p is the pairwise potential.

Unary potential: The unary term measures the cost of assigning label y_i to pixel i and is directly derived from the CNN output probability map:

$$\psi_u(y_i|X)=-log\left(P(y_i|x_i)\right) \quad (2)$$

where $P(y_i/x_i)$ is the softmax probability predicted by the ResNet-50 encoder–decoder for pixel i. A pixel is penalized if its label disagrees with the CNN prediction.

Pairwise potential: The pairwise term encourages nearby and visually similar pixels to take the same label, thereby promoting smoothness and enforcing alignment with image edges. We follow the fully connected CRF model of Krähenbühl and Koltun [23] and use a weighted sum of Gaussian kernels:

$$\psi_p\left(y_i,y_j|X\right)=\mu(y_i,y_j)\sum_{m=1}^{K}w^{(m)}k^{(m)}(f_i,f_j) \quad (3)$$

where $\mu(y_i,y_j)$ =1, if $y_i \neq y_j$ and 0 otherwise (Potts model), $w^{(m)}$ is the weight of the m-th kernel, and $k^{(m)}(f_i, f_j)$ is a Gaussian kernel defined on feature vectors f_i and $f_j$for pixels i and j. We employ a standard bilateral kernel that depends on both spatial position p and RGB color intensity I of the original (non-normalized) image; For CRF inference, we use the original 8-bit RGB image (intensity range [0, 255]) and pixel coordinates on the 256 × 256 grid; the normalization to [0, 1] is applied only to the CNN input and does not affect the CRF potentials.:

$$k\left(f_i,f_j\right)=\omega_1\exp\left(-\frac{\|p_i-p_i\|^2}{2\sigma_\alpha^2}-\frac{\|I_i-I_i\|^2}{2\sigma_\beta^2}\right)+\omega_2\exp\left(-\frac{\|p_i-p_i\|^2}{2\sigma_\gamma^2}\right) \quad (4)$$

The first term connects pixels that are both spatially close and similar in color, encouraging them to share the same label and aligning the segmentation with image edges. The second term is a purely spatial kernel that suppresses small isolated regions and enforces local smoothness. Here, p_i and p_i are measured in pixel units, and I_i and I_i are 8-bit RGB values in $[0,255]^3$.

The CRF hyper-parameters ($\sigma_\alpha,\sigma_\beta,\sigma_\gamma,\omega_1,\omega_2$) are selected via grid search on the validation set, using mean Intersection over Union (mIoU) as the objective. The parameter ranges were chosen to cover the typical scale of spatial and color variations observed in our 256×256 textile images. Specifically, we search $\omega_1\epsilon[1,10]$,*and* $\omega_2\epsilon[1,5]$ with a step size of 1, $\sigma_\alpha\epsilon[50,100]$ with a step size of 10, $\sigma_\beta\epsilon[10,30]$ with a step size of 5, and $\sigma_\gamma\epsilon[1,5]$ with a step size of 1. The best-performing configuration, used in all experiments, is ω_1=5, σ_α=80, σ_β=20, ω_2=3, and σ_γ=3.

Exact minimization of the CRF energy is intractable for a fully connected graph. We therefore employ the efficient mean-field approximation algorithm of Krähenbühl and Koltun [23], as implemented in the publicly available pydensecrf library. This iterative procedure performs approximate inference in linear time with respect to the number of pixels and effectively "cleans up" the CNN probability map, yielding a refined, spatially coherent final segmentation.

EXPERIMENTS AND RESULTS

Experimental Setup

- Hardware: All experiments are conducted on a workstation equipped with an Intel Core i9-12900K CPU, 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of VRAM.

- Software: The deep learning models are implemented in Python 3.8 using the PyTorch framework. The fully connected CRF refinement is performed with the pydensecrf library, a Python wrapper for the original C++ implementation of DenseCRF.
- Training parameters: The ResNet-50 encoder–decoder is trained for 100 epochs with the Adam optimizer, an initial learning rate of 1×10−4, and a batch size of 16. The learning rate is reduced by a factor of 10 every 30 epochs. The same training schedule (optimizer, learning-rate policy, batch size and number of epochs) is applied to all deep segmentation models compared in this study.
- Data augmentation and preprocessing: All image patches are resized to 256×256 pixels, normalized to [0,1], and augmented during training using random horizontal and vertical flips, random rotations in [−15∘,15∘], and slight brightness perturbations. These operations are applied consistently across all neural network models.
- Evaluation metrics: We use four standard metrics for semantic segmentation:
  - Pixel accuracy (PA): The percentage of correctly classified pixels.
  - Mean intersection over union (mIoU): The average IoU over all classes (defect and background). This is our primary metric.
  - Precision:$\frac{TP}{TP+FP}$.
  - Recall: $\frac{TP}{TP+FN}$.
  - F1-Score: The harmonic mean of Precision and Recall, $2\times \frac{Precision\times Recall}{Precision+Recall}$.

Performance Comparison

We compare the proposed Deep-CRF framework with three baseline methods:

- Gabor+SVM: a traditional machine-vision pipeline in which a bank of Gabor filters extracts multi-scale texture features that are classified by a Support Vector Machine.
- U-Net: a widely used encoder–decoder segmentation network with skip connections.
- ResNet-50 (Baseline): .our ResNet-50 encoder–decoder without the CRF post-processing step, trained under the same conditions as Deep-CRF.

Table 2 reports the quantitative results on the STDD test set (300 images).

The conventional Gabor+SVM approach performs worst, with an mIoU of 68.1 % and F1-Score of 67.7 %. It struggles to cope with the heterogeneous textures and subtle defect patterns present in smart textiles.

Both deep learning baselines achieve strong performance: U-Net attains an mIoU of 91.5 % and F1-Score of 92.3 %, while the ResNet-50 baseline reaches an mIoU of 90.2 % and F1-Score of 91.0 %. These results confirm that CNN-based models are well suited for this task.

Our proposed Deep-CRF model achieves the best overall performance, with a PA of 98.2 %, an mIoU of 93.7 %, Precision of 94.5 %, Recall of 93.8 %, and an F1-Score of 94.1 %. Compared to the ResNet-50 baseline, the CRF refinement yields a gain of 3.5 percentage points in mIoU and 3.1 points in F1-Score. It also improves over U-Net by 2.2 points in mIoU and 1.8 points in F1-Score.

To assess the practical feasibility of the methods, we also measure inference latency and peak GPU memory usage. For a 256×256 input patch, U-Net requires 25.1 ms, the ResNet-50 baseline 32.5 ms, and Deep-CRF 45.8 ms. The additional ~13 ms incurred by Deep-CRF is due to the CPU-based DenseCRF post-processing step. Peak GPU VRAM usage of Deep-CRF is identical to that of the ResNet-50 baseline (3.5 GB) because the CRF runs on the CPU. The Gabor+SVM method, executed entirely on the CPU, is considerably slower (≈350 ms per patch). Overall, the extra latency of Deep-CRF is moderate and acceptable for many online quality-inspection scenarios where segmentation accuracy is critical.

Table 2. Performance comparison of different defect detection methods on the STDD test set

Method	PA (%)	mIoU (%)	Precision (%)	Recall (%)	F1 score (%)	Inference time (ms/patch)	Peak GPU VRAM during inference (GB)
Gabor+SVM	85.3	68.1	70.2	65.4	67.7	~350 (CPU)	N/A
U-Net	96.9	91.5	92.1	92.5	92.3	25.1	2.8
ResNet-50 (Base)	97.1	90.2	90.8	91.3	91.0	32.5	3.5
Deep-CRF (Ours)	98.2	93.7	94.5	93.8	94.1	45.8	3.5

Qualitative Analysis

Figure 3 presents representative qualitative results on sample defects from the STDD. For each image, we show the original input, the ground-truth mask, and the output of the Deep-CRF model.

The Deep-CRF predictions closely match the manual annotations, accurately localizing both the extent and shape of the defects. Small spurious regions present in the raw CNN output are largely removed, and defect boundaries are sharper and better aligned with the underlying fabric structures. These qualitative observations are consistent with the quantitative gains reported in Table 2 and highlight the effectiveness of the CRF refinement in producing clean, spatially coherent segmentation masks suitable for downstream inspection and decision making.

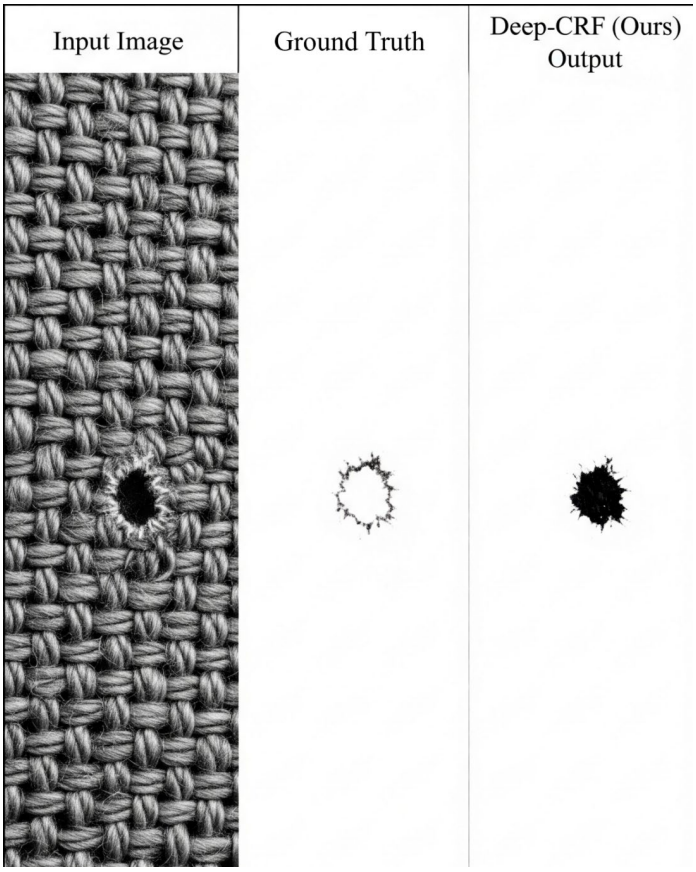

Figure 3. Qualitative visualization of our Deep-CRF segmentation result

Ablation Study

To further analyze the contribution of the main components in our framework, we conduct an ablation study focusing on two factors: data augmentation and CRF refinement. The results are summarized in Table 3.

Effect of data augmentation. When the ResNet-50 encoder–decoder is trained without any data augmentation, its performance on the test set drops to an mIoU of 87.5 % and an F1-Score of 88.1 %. Incorporating the augmentation pipeline described in Section 3.3 improves the mIoU to 90.2 % and the F1-Score to 91.0 %. This confirms that augmentation is crucial for learning robust models from a limited number of annotated smart-textile samples.

Effect of CRF refinement. Adding the fully connected CRF on top of the augmented ResNet-50 baseline further increases the mIoU from 90.2 % to 93.7 % and the F1-Score from 91.0 % to 94.1 %. This substantial improvement demonstrates that the CRF module and the CNN encoder–decoder are complementary: the CNN provides strong local evidence, while the CRF enforces global spatial consistency and corrects residual misclassifications.

Overall, the ablation results show that both data augmentation and CRF refinement are essential to achieving the best performance of the proposed Deep-CRF framework on the STDD.

Table 3. Ablation study of the Deep-CRF framework

Configuration	mIoU (%)	F1 score (%)
ResNet-50 w/o augmentation	87.5	88.1
ResNet-50 w/ augmentation (Baseline)	90.2	91.0
Deep-CRF (Full Model)	93.7	94.1

CONCLUSION

This paper addresses the critical challenge of automated defect detection in smart textile manufacturing. We propose a novel hybrid framework, Deep-CRF, that synergistically combines a deep convolutional neural network (ResNet-50) for powerful feature extraction with a fully-connected conditional random field (CRF) for contextual segmentation refinement.

Our extensive experiments on the newly created Smart Textile Defect Dataset (STDD) demonstrate that Deep-CRF achieves state-of-the-art performance. It significantly outperforms both a traditional machine-vision method and standard deep learning architectures such as U-Net and the ResNet-50 baseline. The integration of the CRF proved highly effective in improving defect localization accuracy, reducing false positives, and generating spatially coherent segmentation masks that are critical for downstream quality assessment.

The proposed system offers a practical and robust solution for enhancing quality control in the smart textile industry. By enabling fast and reliable automated inspection, this technology can help reduce manufacturing costs, improve product reliability, and accelerate the transition of smart textiles from laboratory prototypes to mass-produced consumer goods.

Future work will proceed along several directions. First, we plan to expand the STDD to include more textile types and defect varieties. Second, we will investigate end-to-end trainable models that integrate the CRF directly into the neural network architecture to potentially improve both performance and computational efficiency. Finally, we will explore semi-supervised and unsupervised learning approaches to reduce the manual effort required for pixel-level annotation, making the system easier to adapt to new manufacturing lines.

Author Contributions

Conceptualization – Guang Gao; methodology – Guang Gao and Chuangchuang Chen; investigation – Guang Gao and Chuangchuang Chen; writing-original draft preparation – Guang Gao. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Acknowledgements

Not applicable.

REFERENCES

[1] Stoppa M, Chiolerio A. Wearable Electronics and Smart Textiles: A Critical Review. Sensors. 2014; 14(7):11957-11992. doi: 10.3390/s140711957

[2] Heo JS, Eom J, Kim YH, Park SK. Recent progress of textile-based wearable electronics: A comprehensive review of materials, devices, and applications. Small. 2018; 14(3):1703034. doi: 10.1002/smll.201703034

[3] Cui T-R, Li D, Hirtz T, Shao W-C, Zhou Z-B, Ji S-R, et al. Laser-induced graphene for multifunctional and intelligent wearable systems: For health care and human-computer interaction. Applied Sciences. 2023; 13(8):4688. doi: 10.3390/app13084688

[4] Yin R, Wang D, Zhao S, Lou Z, Shen G. Wearable sensors-enabled human-machine interaction systems: From design to application. Advanced Functional Materials. 2021; 31(11):2008936. doi: 10.1002/adfm.202008936

[5] Castano LM, Flatau AB. Smart fabric sensors and e-textile technologies: A review. Smart Materials and Structures. 2014; 23(5):053001. doi: 10.1088/0964-1726/23/5/053001

[6] Syduzzaman M, Patwary SU, Farhana K, Ahmed S. Smart textiles and nano-technology: A general overview. J. Text. Sci. Eng. 2015; 5(1):1-7.

[7] Kumar A. Computer-vision-based fabric defect detection: A survey. IEEE Transactions on Industrial Electronics. 2008; 55(1):348-363. doi: 10.1109/TIE.1930.896476

[8] Ngan HY, Pang GK, Yung NH. Automated fabric defect detection-A review. Image and Vision Computing. 2011; 29(7):442-458. doi: 10.1016/j.imavis.2011.02.002

[9] Weimer D, Scholz-Reiter B, Shpitalni M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals. 2016; 65(1):417-420. doi: 10.1016/j.cirp.2016.04.072

[10] Ronneberger O, Fischer P, Brox T, editors. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention. Berlin/ Heidelberg, Germany: Springer; 2015. doi: 10.1007/978-3-319-24574-4_28

[11] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems. New York, NY, USA: IEEE; 2015; 28.

[12] Koller D, Friedman N. Probabilistic Graphical Models: Principles and Techniques. Cambridge, MA, USA: MIT Press; 2009.

[13] Haralick RM, Shanmugam K, Dinstein IH. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics. 2007(6): 610-621. doi: 10.1109/TSMC.1973.4309314

[14] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 24(7):971-987. doi: 10.1109/TPAMI.2002.1017623

[15] Tsai D-M, Hsieh C-Y. Automated surface inspection for directional textures. Image and Vision Computing. 1999; 18(1):49-62. doi: 10.1016/S0262-8856(99)00009-8

[16] Cohen FS, Fan Z, Patel MA. Classification of rotated and scaled textured images using Gaussian Markov random field models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 13(2):192-202. doi: 10.1109/34.67648

[17] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems. 2012; 25:84-90. doi: 10.1145/3065386

[18] Redmon J, Divvala S, Girshick R, Farhadi A, editors. You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 27-30 June 2016; Las Vegas, NV, USA; 2016. doi: 10.1109/CVPR.2016.91

[19] Long J, Shelhamer E, Darrell T, editors. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 7-12 June 2015; Boston, MA, USA; 2015. doi: 10.1109/CVPR.2015.7298965

[20] Chen X, Lv J, Fang Y, Du S. Online detection of surface defects based on improved YOLOV3. Sensors. 2022; 22(3):817. doi: 10.3390/s22030817

[21] Wang S, Wu X, Zhang Y, Chen Q. Image crack detection with fully convolutional network based on deep learning. Journal of Computer-Aided Design & Computer Graphics. 2018; 30(5):859-867. doi: 10.3724/SP.J.1089.2018.16322

[22] Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:14127062. 2014.

[23] Krähenbühl P, Koltun V. Efficient inference in fully connected crfs with gaussian edge potentials. Advances in Neural Information Processing Systems. 2011; 24.

[24] He K, Zhang X, Ren S, Sun J, editors. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition; 27-30 June 2016; Las Vegas, NV, USA; 2016. doi: 10.1109/CVPR.2016.90