ICJK
Chinese Journal of Information Fusion
http://dx.doi.org/10.62762/CJIF.2025.864944

RESEARCH ARTICLE

Lightweight Mura Defect Detection via Semantic Interscale Integration and Neighbor Fusion

Zhixi Wang[1,†], **Jinpeng He**[1,†] **and Huaixin Chen**[1,*]

[1] School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China

Abstract

Considering the large-area distribution, smooth brightness gradients, and blurred boundaries of Mura defects in real industrial scenarios, as well as the challenge of balancing accuracy and efficiency in existing methods, we propose a lightweight deep learning-based detection method for large-area Mura defects, termed SIFNet. The SIFNet adopts a classical encoder-decoder architecture with MobileNet-V2 as the backbone. Furthermore, we design a Graph-based Semantic Interscale-fusion Block (GSIB) that integrates the Semantic Fluid Aggregation Module (SFAM) and the Semantic Graph Inference Module (SGIM) to collaboratively extract high-level semantic features across multiple scales and establish abstract semantic representations for accurately localizing large-area Mura defects. Specifically, SFAM leverages a global attention mechanism to extract cross-spatial semantic flows, guiding the model to focus on potential brightness anomaly regions in the image and SGIM explicitly models the semantic relationships between multi-scale features using graph convolution, enhancing the model's ability to interpret regions with blurred boundaries and ambiguous structures. To further improve the model's sensitivity to edges in regions with smooth brightness transitions, we introduce a NeighborFusion Edge Enhancement Module (NEEM). This module integrates depthwise separable convolutions with a spatial attention mechanism and introduces a CrossNorm-based feature alignment strategy to enhance spatial collaboration across feature layers. Additionally, an edge enhancement mechanism is employed to significantly improve the model's ability to delineate blurred Mura defect boundaries, while keeping computational cost low and strengthening edge perception and representation. Extensive quantitative and qualitative experiments on three large-area Mura defect datasets constructed in this study demonstrate that SIFNet achieves excellent detection performance with only 3.92M parameters and 6.89 GFLOPs, striking an effective balance between accuracy and efficiency, and fully meeting the demands of industrial deployment.

Keywords: mura defect detection, lightweight network, interscale-fusion, neighbor fusion.

Academic Editor:
You He

Submitted: 25 April 2025
Accepted: 23 July 2025
Published: 25 September 2025

Vol. 2, **No.** 3, 2025.
10.62762/CJIF.2025.864944

***Corresponding author:**
✉ Huaixin Chen
huaixinchen@uestc.edu.cn

† These authors contributed equally to this work

Citation
Wang, Z., He, J., & Chen, H. (2025). Lightweight Mura Defect Detection via Semantic Interscale Integration and Neighbor Fusion. *Chinese Journal of Information Fusion*, 2(3), 237–252.

1 Introduction

As a core component of modern electronic devices, the display panel plays a critical role in shaping the end-user's visual experience. However, due to inherent limitations in material properties, manufacturing techniques, and production environments, surface defects on display screens remain difficult to avoid during the fabrication process [1]. Among the various display quality defects, Mura defects, which refer to brightness non-uniformity, are particularly prominent. They often cause visible brightness variations on the screen and significantly impair the overall visual experience [2]. Unlike local defects such as dot or line defects, Mura defects typically span large areas, exhibit complex and diverse patterns, and involve subtle brightness variations, posing significant challenges for traditional detection methods in terms of both identification and localization. Moreover, this brightness non-uniformity not only affects visual performance but may also reduce the lifespan of the display. Therefore, accurately detecting and effectively segmenting non-uniform regions on the screen has become a critical issue that demands urgent attention [3].

Traditional Mura defect detection methods are primarily based on image processing techniques, including frequency domain analysis [4, 20], filtering and statistical analysis [5, 6], and low-rank matrix decomposition. Frequency domain approaches, such as Fourier transform and discrete cosine transform, achieve defect segmentation by removing background textures. However, their high computational cost in background reconstruction limits their applicability in real-time scenarios. Filtering and statistical methods rely on techniques like mean filtering [7] and Gaussian filtering [8] to highlight non-uniform regions, followed by threshold-based segmentation. However, these methods are prone to interference from local features when handling complex or subtle defects and are highly sensitive to parameter settings, resulting in limited adaptability. Low-rank matrix decomposition models the background as a low-rank matrix and treats defects as sparse components, separating the two via optimization algorithms. Although this approach shows promising performance in Mura defect detection, its high computational complexity similarly restricts its use in real-time industrial applications. Overall, while traditional methods may be effective in specific scenarios, they generally suffer from strong dependence on background regularity, high computational costs, and poor generalization, making them insufficient for the diverse demands of Mura defect detection in practical industrial settings [9].

In recent years, the rapid advancement of display technology has led to high-resolution and large-size panels becoming the market mainstream, imposing greater demands on large-area Mura defect detection. Higher resolutions require algorithms to process significantly larger volumes of data, while increased panel sizes add complexity to the detection task. At the same time, the rise of deep learning has introduced new perspectives for tackling Mura defects. Detection methods based on Convolutional Neural Networks (CNNs) have become the dominant approach, offering efficient feature extraction and semantic segmentation through end-to-end training frameworks. Significant progress has been made in applying deep learning to LCD defect detection, with three main research directions emerging: (1) Optimization and application of enhancements of object detection model to classical architectures such as YOLO [10] and RetinaNet [11], combined with attention mechanisms, have improved performance in complex backgrounds and for various defect types; (2) Addressing the scarcity of labeled data through transfer learning and unsupervised learning approaches, reducing the reliance on large-scale annotated datasets and improving model adaptability [12, 13]; (3) Combining lightweight network designs with data augmentation strategies to reduce computational overhead and enable real-time detection in resource-constrained environments [14]. Nevertheless, existing object detection networks remain limited in their ability to handle large-area Mura defects with blurred boundaries, and often struggle to balance detection accuracy with the stringent real-time requirements of industrial settings, as illustrated in Figure 1.

Moreover, compared to conventional display defect detection, Mura defect detection presents distinct differences in both research focus and technical requirements. It emphasizes the overall brightness distribution across the screen rather than the specific shapes of localized defects. As a result, the task requires capturing large-scale brightness gradients and non-uniformities, which often lack clear boundaries or geometric features. Furthermore, Mura defects typically exhibit diffuse patterns and are influenced by complex factors such as backlighting, display environment, and screen content, making general object detection methods difficult to apply effectively. Large-area Mura defect detection faces several key

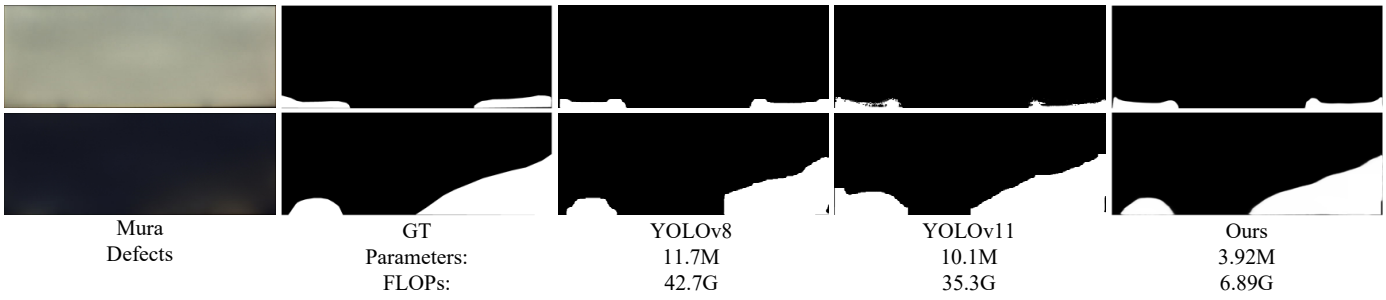

Figure 1. A visual comparison between our method and commonly adopted object detection methods (YOLOv11 [15], YOLOv8 [16]) in industrial applications.

challenges: (1) Mura defects often appear as subtle brightness gradients or broad regions of uneven illumination, making them difficult to differentiate from the background; (2) These defects generally lack sharp boundaries or distinct structural cues, complicating accurate localization; (3) Limited availability of annotated defect samples hinders model training and restricts generalization capabilities; (4) Processing high-resolution images demands greater computational efficiency and detection accuracy, requiring methods that balance precision with real-time performance.

To address the above challenges, we focus on developing a lightweight deep learning-based method for large-area Mura defect detection. We propose a novel lightweight Semantic Interscale Integration and Neighbor Fusion-based network, termed SIFNet, which aims to enhance Mura defect detection performance while maintaining a lightweight design suitable for industrial applications. Specifically, we adopt a commonly used encoder-decoder structure with MobileNet-V2 [17] as the backbone. To effectively capture large-area Mura defects and enhance the model's sensitivity to subtle brightness variations, we design a Graph-based Semantic Interscale Fusion Block (GSIB). This block integrates high-level semantic information across spatial scales and models global feature relationships within Mura defect regions, while maintaining low computational complexity. GSIB consists of two submodules: the Semantic Fluid Aggregation Module (SFAM) and the Semantic Graph Inference Module (SGIM). SFAM adopts a holistic strategy to extract global semantic cues, while SGIM employs graph convolution operations to explicitly model cross-scale semantic relationships by constructing higher-order connections between layers. To further improve the model's ability to detect fine boundary features in regions of brightness non-uniformity, we introduce the NeighborFusion Edge Enhancement Module (NEEM). NEEM enhances edge representation by mining spatial correlations between multilevel features. Finally, to address the diverse scales and morphologies of Mura defects, we design a lightweight multiscale decoder for accurate detection across varying defect sizes. In summary, our main contributions are as follows:

1. We propose a novel lightweight network, SIFNet, for large-area Mura defect detection, and construct three high-resolution large-area Mura defect datasets to support model training and evaluation.

2. We design a lightweight Graph-based Semantic Interscale-fusion Block (GSIB) to effectively capture large-area Mura defects and subtle brightness variations. GSIB integrates multi-level deep semantic features and models cross-scale relationships through two components: SFAM, which extracts holistic semantic cues, and SGIM, which builds high-order dependencies via graph convolution.

3. We propose a lightweight NeighborFusion Edge Enhancement Module (NEEM) that strengthens spatial correlations among multi-level features and reinforces edge representations, thereby improving the model's sensitivity to subtle boundary features in Mura defect regions.

4. Comprehensive experiments on the proposed datasets demonstrate that our method achieves state-of-the-art performance while being lightweight, highlighting its suitability for industrial applications.

2 Related Work

2.1 Traditional image processing-based Mura defect detection methods

Early traditional defect detection methods primarily relied on thresholding, edge information, feature

extractors, and classifiers for defect classification and localization. Sun et al. [18] proposed a cascaded Mura detection approach based on mean shift and level set algorithms, where the mean shift algorithm is first used to locate potential defect regions, followed by segmentation using the level set method. Chen et al. [19] addressed the challenge of detecting Mura defects with low brightness contrast against the background by proposing a background reconstruction method based on discrete cosine transform (DCT), generating a background image separated from the defect. They further quantified the saliency of Mura defects using the Just Noticeable Difference (JND) model. Tsai et al. [20] tackled the difficulties of detecting defects in large-size, high-density LCDs with complex and irregular patterns by applying Fourier-based image reconstruction to remove periodic backgrounds from 1D line patterns, enabling accurate segmentation of local anomalies. Cui et al. [21] employed three visual-based techniques for display defect detection and used OTSU thresholding to segment multiple defect types across various uniform background colors. Chen et al. [22] proposed a salt-and-pepper defect detection method using mean filtering and statistical control charts. The method first identifies defect pixels via mean filtering and binarization, then monitors the number of detected pixels with a control chart to determine panel-level defects. However, these traditional methods typically rely on background regularity, involve high computational costs, and have limited adaptability in complex or variable scenarios. Moreover, their accuracy is often unstable, making them unsuitable for direct deployment on industrial production lines.

2.2 Deep learning-based Mura defect detection methods

With the rapid development of deep learning, many methods have adopted end-to-end training frameworks that enable efficient extraction of display defect features and semantic segmentation. In the field of LCD defect detection, deep learning-based approaches have made notable progress, with research primarily centered on the optimization and application of general object detection models. Zhu et al. [23] employed YOLOv3 as the detection framework to identify defects and noise under varying backgrounds and viewing angles. By subtracting the noise detection results from the defect results, the method achieved a more accurate defect localization. Çelik et al. [24] analyzed deep learning-based LCD defect detectors using RetinaNet, YOLOv3, and M2Det, and found that RetinaNet offered the best balance between accuracy and processing time. Luo et al. [9] improved the YOLOv5 backbone by widening both shallow and deep convolutional layers, and proposed a shallow semantic fusion module to enhance the utilization of low-level features. They also introduced a contrast-enhancing attention mechanism to strengthen feature representations across spatial and channel dimensions.

Furthermore, extensive efforts have been made to design efficient neural network architectures that reduce computational overhead while improving detection accuracy. Lin et al. [25] used MobileNet as a baseline and incorporated a channel attention mechanism to enhance channel-wise features for Mura defect detection in LCDs. Chen et al. [14], building upon the YOLOv4-tiny framework, integrated atrous spatial pyramid pooling and depthwise separable convolutions to expand the receptive field, while spatial and channel attention modules were applied to further enhance feature maps. These improvements enable faster and more precise defect detection, supporting real-time applications in resource-constrained environments.

3 Method

In this section, we provide a detailed description of the proposed lightweight SIFNet. Section 3.1 presents an overview of the overall architecture. Sections 3.2 and 3.3 explain the two lightweight modules, GSIB and NEEM, respectively. Finally, Section 3.4 describes the decoder structure and the loss function used for training.

3.1 Overview

As shown in Figure 2, the proposed lightweight SIFNet is built upon a common encoder-decoder architecture [29, 30] and consists of six key components: Mv2 encoder, Graph-based Semantic Interscale-fusion Block (GSIB), Semantic Fluid Aggregation Module (SFAM), Semantic Graph Inference Module (SGIM), NeighborFusion Edge Enhancement Module (NEEM), and lightweight decoder. For the encoder of SIFNet, we adopt the lightweight MobileNet-V2 [17]. Specifically, we retain the first 17 inverted residual bottlenecks while truncating the last two convolutional layers and average pooling layer to make it suitable for Mura detects detection. We divide MobileNet-V2 into five blocks based on the first, third, sixth, 13th, and last bottlenecks. The output five-level features are denoted as $F_i \in R^{c_i \times h_i \times w_i}$, i = 1,2...5, where h_i and

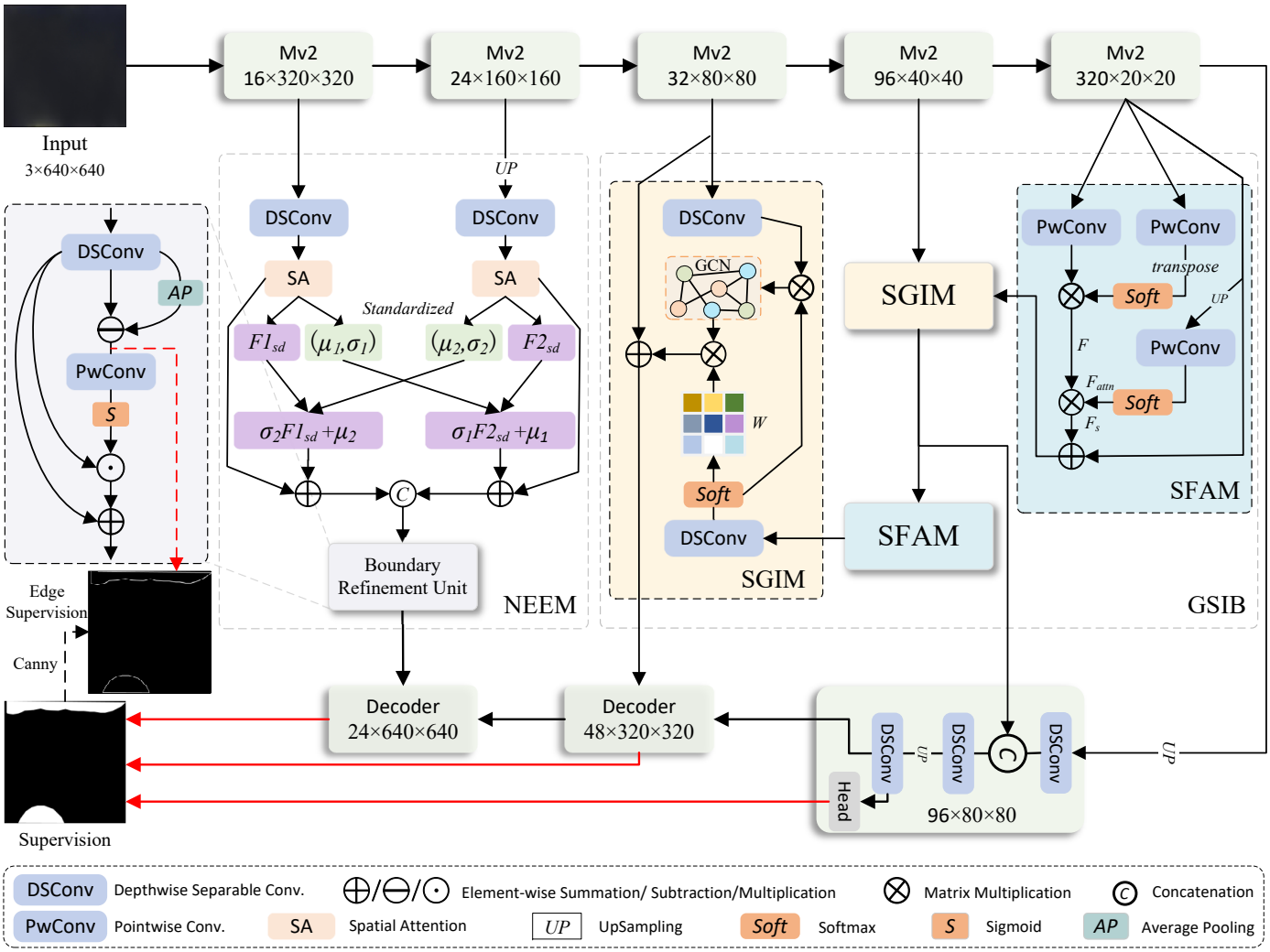

Figure 2. Overall architecture of the proposed SIFNet. It consists of six key components: Mv2 encoder, Graph-based Semantic Interscale-fusion Block (GSIB), Semantic Fluid Aggregation Module (SFAM), Semantic Graph Inference Module (SGIM), NeighborFusion Edge Enhancement Module (NEEM), and lightweight decoder.

w_i are $(640/2^i)$, and $c_i \in \{16, 24, 32, 96, 320\}$. Next, to fuse multi high-level semantic features and model abstract semantic concepts for more precise defect localization, the features $\{F_3, F_4, F_5\}$ are processed by the GSIB, which integrates SFAM and SGIM to extract global contextual information and aggregate cross-scale semantics via hierarchical flow fusion and graph convolution. In addition, NEEM explores the correlation between shallow features $\{F_1, F_2\}$, enhancing the model's ability to perceive fine-grained structures and represent spatial details more effectively. Furthermore, edge information is extracted using a pooling subtraction operation [31], and the Canny [32] operator is applied to generate edge supervision for correcting edge errors. Finally, three decoders are employed in a progressive manner to emphasize target objects and restore resolution.

3.2 Graph-based Semantic Interscale-fusion Block

Mura defects typically exhibit large-area distribution, accompanied by uneven brightness variations and blurred boundaries, making precise segmentation challenging. To address these issues, we propose the Graph-based Semantic Interscale Fusion Block (GSIB), which integrates high-level semantic information across spatial scales and efficiently extracts semantic cues from the spatial context. By combining graph-based modeling with a soft attention mechanism, GSIB excels at capturing long-range dependencies and maintaining global semantic consistency. It effectively activates high-level features in potential defect regions and enhances synergy between spatial location and semantic representation, enabling accurate modeling of global feature relationships within Mura-affected areas. GSIB improves the model's ability to perceive

large-scale brightness gradients while maintaining low computational complexity and parameter overhead, achieving a strong balance between performance and efficiency. As shown in Figure 2, GSIB comprises two key components: the Semantic Fluid Aggregation Module (SFAM) and the Semantic Graph Inference Module (SGIM).

(1) Semantic Fluid Aggregation Module. SFAM employs a holistic strategy to capture global semantic features, enabling comprehensive extraction of semantic cues across the entire image. As shown in Figure 2, taking the encoder feature $F_5 \in \mathbb{R}^{C\times H\times W}$ as an example, SFAM introduces a global attention mechanism to selectively extract visual primitives along the spatial dimension, thereby generating a semantic flow representation F. Specifically, F_5 is passed through two parallel 1×1 convolutional branches to produce a feature query $F_{\mathrm{q}} \in \mathbb{R}^{C\times HW}$ and attention weights $F_\omega \in \mathbb{R}^{HW\times N}$. The attention branch applies a softmax function to normalize the spatial dimension, resulting in a saliency distribution across the spatial domain. Finally, matrix multiplication is used to integrate the attention-guided global semantic representation, producing the semantic flow feature $F \in \mathbb{R}^{C\times N}$:

$$F_{\mathrm{q}} = \varphi(F; W_\varphi) \in \mathbb{R}^{C\times HW}, \tag{1}$$

$$F_\omega^\top = \mathrm{softmax}(\theta(F; W_\theta))^\top \in \mathbb{R}^{HW\times N}, \tag{2}$$

$$F = F_{\mathrm{q}} \otimes F_\omega^\top, \tag{3}$$

where $\otimes$ denotes matrix multiplication. φ and θ represent two independent 1×1 convolution operations, with W_φ and W_θ denoting their respective convolution parameters.

Due to the semantic gap between high-level features at different scales, direct multi-scale fusion may amplify inconsistencies between adjacent layers. To address this issue, the semantic flow F is further refined using a soft attention mechanism that selectively emphasizes informative input features before passing them to the SGIM for interscale fusion. Specifically, the input feature F_5 is upsampled and processed through a 1×1 convolution, followed by a softmax operation along the channel dimension, producing an N-dimensional attention map $F_{\mathrm{att}} \in \mathbb{R}^{N\times kH\times kW}$, where $k = 2$ in our experiments. Guided by this attention map, the semantic flow $F \in \mathbb{R}^{C\times N}$ is adaptively integrated into the spatial domain, resulting in a semantic description map $F_s \in \mathbb{R}^{C\times kH\times kW}$. Finally, the semantic description map F_s is fused with the original high-level feature F_5 via element-wise addition to obtain the final output of SFAM, denoted as $F_{sfam} \in \mathbb{R}^{C\times kH\times kW}$. The computation is formally defined as:

$$F_{\mathrm{att}} = \mathrm{softmax}(\phi(UP(F); W_\phi)), \tag{4}$$

$$F_s = F \otimes F_{\mathrm{att}}, \tag{5}$$

$$F_{sfam} = F_s + F_5, \tag{6}$$

where $UP(\cdot)$ denotes the upsampling operation, and ϕ represents the 1×1 convolution.

In summary, SFAM effectively enhances global context understanding by integrating semantic flow with high-level features, while the use of soft attention helps bridge the semantic gap between features at different scales, providing a solid foundation for subsequent feature fusion in SGIM.

(2) Semantic graph-inference module. When dealing with complex Mura defect regions, traditional feature fusion methods often struggle to effectively capture the diversity of brightness variations and the inherent structural ambiguity. Therefore, we propose the Semantic Graph Inference Module (SGIM), which introduces an efficient graph-based feature aggregation mechanism to integrate receptive field information across multiple feature levels. Specifically, SGIM constructs high-order interscale relationships within the graph convolution domain, enabling explicit modeling of semantic dependencies through graph convolution operations. For Mura defects with smooth brightness transitions and blurred boundaries, SGIM effectively captures global brightness variation patterns across spatial regions, significantly improving both detection accuracy and robustness. Additionally, SGIM employs depthwise separable convolution blocks to reduce computational overhead. The detailed architecture is illustrated in Figure 2.

Taking the example shown in Figure 2, the SFAM output feature F_{sfam} and encoder feature F_3 are first processed through two learnable linear mapping functions, φ_1 and φ_2, producing spatially aligned feature sequences S_1 and S_2, respectively: $S_1 = \varphi_1(F_{sfam}), \quad S_2 = \varphi_2(F_3)$.

Next, a softmax function is then applied to S_1 to generate the attention map W: $W = \mathrm{Softmax}(S_1)$.

The attention map is then multiplied with S_2, and the resulting features are fed into a Graph Convolutional Network (GCN) [39] to learn high-order semantic relationships among regions (sets of pixels with similar features). To reconstruct the graph domain features back into their original structural features, the inner

product of GCN$(\cdot)$ is computed and transformed back into a 2D image feature map with the same dimensions as the original features using a linear mapping function ϕ_f. Finally, the reconstructed features are then combined with feature F_3 to produce the final output of SGIM, F_{sgim}, which is then passed to SFAM (if present) and the decoder. Notably, the number of nodes in the GCN is set to 16.

$$F_{sgim} = \varphi_f(W^T \otimes \text{GCN}(W^T \otimes S_2)) + F_3. \quad (7)$$

3.3 NeighborFusion Edge Enhancement Module

Shallow features contain rich texture details and edge information, which are critical for accurately delineating Mura defect boundaries, but they also introduce considerable background noise. Moreover, Mura defects typically exhibit gradual brightness transitions characterized by smooth intensity variations and indistinct boundaries without clear geometric structures. These features pose significant challenges for most existing defect detection methods. Since Mura defects typically exhibit gradual brightness transitions, their smooth intensity changes and blurred boundary characteristics often lack discernible geometric structures, making such regions particularly challenging for most methods to handle effectively. Thus, we propose the NeighborFusion Edge Enhancement Module (NEEM), which is designed to capture spatial correlations across multilevel features and enhance the model's sensitivity to boundary features in regions with uneven brightness. This enables more accurate identification of Mura defects in complex display environments. Additionally, NEEM is lightweight and computationally efficient, providing a practical solution for large-area Mura defect detection. The specific implementation details are illustrated in Figure 2.

First, the encoder features F_2 and F_1 are spatially aligned through upsampling. Depthwise separable convolutions (DSConv) are then applied to adjust their channel dimensions to match those of F_2, with spatial attention independently applied to each feature to enhance detail representation and prepare for subsequent fusion. The resulting feature maps are denoted as F_1' and $F_2' \in \mathbb{R}^{24\times320\times320}$.

To facilitate information interaction between different feature levels, CrossNorm [40] is employed to compute and exchange the mean and variance of F_1' and F_2', enabling structured context transfer and improving the perception of fine-grained structures. Finally, the two refined features are fused via concatenation, and a residual connection is introduced to generate the final enhanced representation, improving the consistency of edge depiction and semantic expression. This process can be expressed as follows:

$$F_1^{corr} = F_1' + CrossNorm(F_1', F_2'), \quad (8)$$
$$F_2^{corr} = F_2' + CrossNorm(F_1', F_2'), \quad (9)$$
$$F^{corr} = cat(F_1^{corr}, F_2^{corr}), \quad (10)$$

where $cat(\cdot)$ indicates channel-wise concatenation.

Next, a boundary enhancement module is applied to further refine the representation of edge features in Mura defect regions, generating the NEEM output F_{neem}, which is subsequently passed to the decoder. This process is defined as follows:

$$F_g = DSConv(F^{corr}), \quad (11)$$
$$f_e = F_g \ominus \text{AvgPool}(F_g), \quad (12)$$
$$F_{neem} = \text{Sigmoid}(\text{PWConv}(f_e)) \odot F_g \oplus F_g, \quad (13)$$

where Sigmoid$(\cdot)$ denotes the activation function, and $\oplus$ represents element-wise addition.

3.4 Decoder and Loss Function

Mura defects often exhibit diverse scales and brightness variations across different regions of a display, necessitating precise feature fusion at multiple levels. Therefore, we introduce a lightweight multi-scale decoder that adaptively integrates hierarchical features while progressively restoring spatial resolution. This approach enables accurate detection of Mura defects across varying scales, ensuring reliable localization and classification of both small, subtle imperfections and large-area brightness non-uniformities. Each decoder follows a consistent architecture, as illustrated in Figure 2. Specifically, each decoder sequentially consists of a DSConv layer, a Concate layer (if present, to merge NEEM or SGIM outputs), another DSConv layer, an upsampling layer, a final DSConv layer, and a Head layer. The final DSConv layer provides the decoder's output for the next stage, while the Head layer generates the prediction output for deep supervision. Specifically, the Head layer consists of a dropout layer and a 1×1 convolutional layer, which generate three saliency maps with different resolutions: $f_3^{pre} \in [0,1]^{1\times80\times80}$, $f_2^{pre} \in [0,1]^{1\times320\times320}$, and $f_1^{pre} \in [0,1]^{1\times640\times640}$. The first two are used for deep supervision, while the last serves as the final output of SIFNet.

After obtaining the saliency maps $\{f_i^{pre}\}, i = 1,2,3$ and the edge prediction map $\{f_e^{pre}\}$, we train our

SIFNet using a combination of BCE loss and IoU loss [33]. Therefore, our total loss consists of two parts, that is, the saliency loss and the edge loss. Moreover, we introduce a loss weight to balance the two losses, enabling more effective training. The total loss function L_{total} can be formulated as follows:

$$L_{total} = L_{bce}(f_e^{pre}, G_e) + \sum_{i=1}^{3} \frac{1}{2^{i-1}} \left(L_{bce}(f_i^{pre}, G) + L_{iou}(f_i^{pre}, G)\right) \tag{14}$$

where L_{bce} and L_{iou} represent the BCE loss and IoU loss, respectively, which are used to supervise f_i^{pre} with the ground truth G. G_e denotes the edge supervision map, generated from G using the Canny operator.

4 Experiments

4.1 Experiment Settings

Implementation Details. We use MobileNetV2 as the encoder, pre-trained on the ImageNet dataset [34], and the other modules are randomly initialized. For training and testing, all input images are resized to 640×640, and random flipping and rotation are applied to augment the training data. We use Adam as the optimizer, with the learning rate initialized to 1e-4 and then scaled down by 10 every 30 epochs. Our model is trained end-to-end using PyTorch on an NVIDIA A100-SXM for 100 epochs with a batch size of 8.

Datasets. Given the limited availability of Mura defect samples from display screens, we have constructed three specialized datasets for Mura defect detection: the black screen scenario (Uneven_Black), the white screen scenario (Uneven_White), and a mixed black-and-white screen scenario (Uneven_Mix). All images have a resolution of 3612×1358 pixels. The Uneven_Black dataset consists of 350 images exhibiting uneven displays on a black background, with 245 images allocated for training and 105 for testing. The Uneven_White dataset also comprises 350 images captured on a white background, following the same training/testing split. The Uneven_Mix dataset combines scenes from both black and white backgrounds, comprising 700 images in total, with 490 used for training and 210 for testing. Representative image samples are illustrated in Figure 3.

Evaluation Metrics. To comprehensively evaluate the performance of the model, we performed an extensive evaluation from both the object-level and pixel-level perspectives. The details of these evaluation metrics

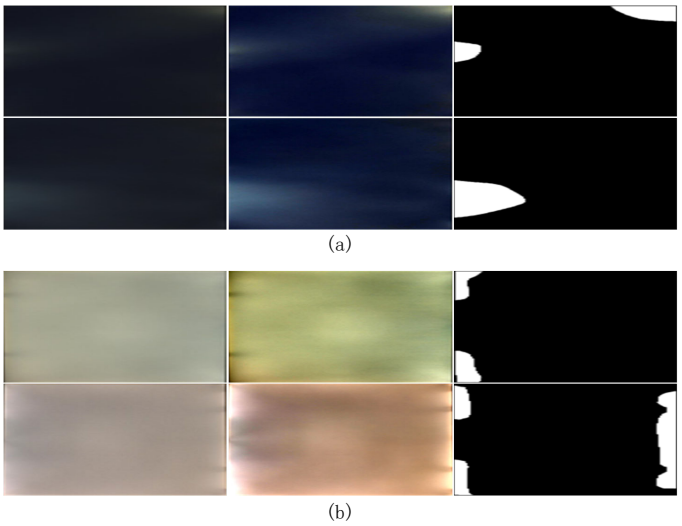

Figure 3. Examples of display defect under black and white screen conditions. (a) the black background scenario, (b) the white background scenario. The images from left to right are the original image, the contrast-enhanced image, and the corresponding ground truth mask, respectively.

are as follows:

(1) Object-level:

Precision and Recall. In object detection, precision refers to the proportion of correctly predicted positive samples among all samples predicted as positive, while recall represents the proportion of correctly predicted positive samples among all actual positive samples. They are defined as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{15}$$

$$Recall = \frac{TP}{TP + FN}, \tag{16}$$

where TP (True Positive) refers to correctly detected objects, FP (False Positive) denotes incorrectly identified objects, and FN (False Negative) indicates actual objects that were missed by the model.

(2) Pixel-level:

Precision and Recall. In pixel-level detection, precision is defined as the ratio of correctly predicted regions to the total predicted regions, while recall is defined as the ratio of correctly predicted regions to the total ground truth regions. The formal definitions are as follows:

$$Precision = \frac{|S(T) \cap G|}{S(T)}, \tag{17}$$

$$Recall = \frac{|S(T) \cap G|}{G}, \tag{18}$$

where $S(T)$ represents the predicted object region by the model and G denotes the ground truth object region.

S-measure [35] is used to evaluate region-aware(S_r) and object-aware(S_o) structural similarity between predictions and GT and defined as:

$$S = \alpha S_o + (1 - \alpha) S_r, \tag{19}$$

where α is set to 0.5.

F-measure [36] is a holistic metric that considers both precision (P) and recall (R), which is defined as:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{20}$$

where β is the balance parameter and β^2 is set as 0.3.

E-measure [37] is used to measure pixel-level matching and image-level statistics, which is defined as:

$$E = \frac{1}{N} \sum_{i=1}^{N} \phi FM(i), \tag{21}$$

where ϕFM denotes the enhanced-alignment matrix and N is the total pixels of the image.

Mean absolute error [38] is to calculate the average absolute error of the prediction of salient objects (P) and ground truth (G), which is defined as:

$$M = \frac{1}{N} \sum_{i=1}^{N} |P(i) - G(i)|, \tag{22}$$

4.2 Comparison with State-of-the-Art Methods

We conducted a systematic comparative analysis of recent visual models that have shown strong performance in object detection and instance segmentation tasks. In this study, we selected five representative versions from the YOLO series: YOLOv7 [26], YOLOv8 [16], YOLOv9 [27], YOLOv10 [28], and YOLOv11 [15] as our research targets. These models are built on advanced deep learning architectures and significantly improve detection accuracy and segmentation quality while maintaining high real-time performance. They have been widely applied in industrial settings and defect detection tasks, especially in scenarios that require both speed and precision, such as display panel inspection and surface quality analysis. We carried out a comprehensive evaluation of these models across several aspects, including object-level and pixel-level performance, computational efficiency, and generalization ability.

Quantitative Results. Tables 1, 2, and 3 summarize the quantitative results of our proposed method and competing methods on the three challenging large-area Mura defect datasets. As can be seen from the results:

- On the Uneven_Black dataset (Table 1), our method achieves the highest scores in all object-level and pixel-level metrics. It outperforms all baseline models with a precision of 0.839 and a recall of 0.865 at the object level, and leads in pixel-level evaluation with strong results in structural similarity, alignment quality, and minimal error (M = 0.067). These results highlight our model's superior ability to detect Mura defects under low-contrast, dark background conditions.
- On the Uneven_White dataset (Table 2), our method again delivers the best overall performance. It achieves the highest object-level precision and recall (0.912 and 0.906, respectively) and leads in key pixel-level metrics such as F-measure (F_m^{mean} = 0.626). Our method also attains a very low error rate (M = 0.029), confirming its robustness in handling subtle defects under high-brightness settings.
- On the Uneven_Mix dataset (Table 3), which combines both dark and bright regions, our model consistently achieves top performance. It records the highest object-level precision (0.859) and recall (0.823), and outperforms all baselines across pixel-level metrics. These results demonstrate that the proposed SIFNet not only performs well on specialized datasets, but also generalizes effectively to more diverse and complex industrial defect scenarios.

Qualitative results. In the Mura defect detection task, accurately segmenting defective regions is critical for improving quality control in display panel manufacturing. Figures 4 and 5 illustrate the detection results under two different background conditions (black screen and white screen), comparing various methods. As can be seen from the results:

- Under the black screen scenario, YOLOv11 and YOLOv7 generally capture the overall shape of the defects but suffer from boundary inaccuracies and occasional over- or under-segmentation. YOLOv8 and YOLOv10 perform less effectively, producing coarser segmentations and distorted defect shapes.

Table 1. Performance comparison of different methods on the 'Uneven_Black' dataset. The best and second performing methods of each category are highlighted in **bold** and underlined, respectively.

Methods	#Param (M)	FLOPs (G)	Object-level		Pixel-level					
			P	R	P	R	S_m	E_m^{mean}	F_m^{mean}	M
YOLOv7	37.8	141.9	0.789	0.665	0.779	0.576	0.692	0.815	0.690	0.074
YOLOv8	11.7	42.7	0.783	0.605	0.787	0.497	0.663	0.782	0.645	0.080
YOLOv9	57.8	371.6	0.753	0.684	0.704	0.631	0.672	0.815	0.677	0.083
YOLOv10	8.7	21.6	0.750	0.674	0.748	0.716	0.664	0.744	0.668	0.086
YOLOv11	10.1	35.3	0.722	0.780	0.727	0.736	0.685	**0.855**	0.699	0.069
Ours	3.92	6.89	**0.839**	**0.865**	**0.800**	**0.841**	**0.704**	0.832	**0.742**	**0.067**

Table 2. Performance comparison of different methods on the 'Uneven_White' dataset. The best and second performing methods of each category are highlighted in **bold** and underlined, respectively.

Methods	#Param (M)	FLOPs (G)	Object-level		Pixel-level					
			P	R	P	R	S_m	E_m^{mean}	F_m^{mean}	M
YOLOv7	37.8	141.9	0.814	0.809	0.820	0.755	0.672	0.578	0.096	0.043
YOLOv8	11.7	42.7	0.891	0.512	0.876	0.464	0.781	0.892	0.456	0.026
YOLOv9	57.8	371.6	0.819	0.823	0.859	**0.804**	0.673	0.569	0.076	0.043
YOLOv10	8.7	21.6	0.856	0.785	0.811	0.768	0.738	0.682	0.087	0.029
YOLOv11	10.1	35.3	0.889	0.548	0.871	0.725	0.794	0.904	0.464	**0.024**
Ours	3.92	6.89	**0.912**	**0.906**	**0.895**	0.803	**0.890**	**0.965**	**0.626**	0.029

Table 3. Performance comparison of different methods on the 'Uneven_Mix' dataset. The best and second performing methods of each category are highlighted in **bold** and underlined, respectively.

Methods	#Param (M)	FLOPs (G)	Object-level		Pixel-level					
			P	R	P	R	S_m	E_m^{mean}	F_m^{mean}	M
YOLOv7	37.8	141.9	0.799	0.775	0.780	0.735	0.682	0.696	0.394	0.058
YOLOv8	11.7	42.7	0.825	0.571	0.818	0.565	0.722	0.837	0.551	0.053
YOLOv9	57.8	371.6	0.778	0.755	0.796	0.725	0.673	0.693	0.376	0.063
YOLOv10	8.7	21.6	0.807	0.723	0.782	0.746	0.691	0.714	0.412	0.057
YOLOv11	10.1	35.3	0.787	0.735	0.821	0.731	0.740	0.879	0.581	0.047
Ours	3.92	6.89	**0.859**	**0.823**	**0.824**	**0.831**	**0.803**	**0.902**	**0.632**	**0.042**

Table 4. Ablation results of evaluating the contribution of each module in SIFNet on the Uneven_Mix dataset. The best one in each column is **Bold**.

Methods	FPS	#Param (M)	FLOPs (G)	Object-level		Pixel-level					
				P	R	P	R	S_m	E_m^{mean}	F_m^{mean}	M
No.1	61	2.355	2.58	0.794	0.773	0.761	0.777	0.740	0.840	0.561	0.112
No.2	57	2.781	3.46	0.818	0.792	0.789	0.791	0.766	0.887	0.592	0.083
No.3	50	3.567	5.97	0.831	0.804	0.802	0.811	0.782	**0.908**	0.611	0.059
No.4	45	3.922	6.89	**0.859**	**0.823**	**0.824**	**0.831**	**0.803**	0.902	**0.632**	**0.042**

Figure 4. Visual comparison of Mura defect detection results under the black screen scenario.

Figure 5. Visual comparison of Mura defect detection results under the white screen scenario.

Table 5. Performance comparison of SIFNet on the Uneven_Black and Black_aug datasets.

Datasets	P	R	$S_m \uparrow$	$F_\beta^{max} \uparrow$	$F_\beta^{mean} \uparrow$	$F_\beta^{adp} \uparrow$	$E_\phi^{max} \uparrow$	$E_\phi^{mean} \uparrow$	$E_\phi^{adp} \uparrow$	$M \downarrow$
Uneven_Black	0.800	0.841	0.704	0.832	0.742	0.852	0.748	0.846	0.748	0.067
Black_aug	0.851	0.961	0.875	0.976	0.901	0.984	0.924	0.987	0.903	0.010

In particular, YOLOv11 is prone to false positives, often misclassifying background artifacts as defects. On the contrast, SIFNet exhibits a strong ability to delineate defect boundaries and closely reconstruct the true shapes of the defective areas. Its predictions are highly consistent with the GT, characterized by smooth contours and precise coverage.

- Under the white screen scenario, the reduced contrast between defects and background makes detection more challenging. SIFNet continues to perform reliably, producing accurate results that closely align with the ground truth and effectively retain fine boundary details. In comparison, the competitors show reduced performance: YOLOv11 and YOLOv8 often yield blurrier edges and miss subtle defect regions, while YOLOv9 and YOLOv10 exhibit more pronounced issues, such as false positives, incomplete segmentation, and disrupted boundaries, indicating limited generalization and detail preservation under complex scenarios.

Overall, unlike other methods that exhibit inconsistent performance across different background scenarios, SIFNet demonstrates consistently strong results, accurately capturing Mura defect features with strong robustness.

4.3 Ablation Studies.

To analyze the contribution of each lightweight module in our approach to Mura defect detection, we evaluate four variants of the model: No.1. Baseline (consists of the MobileNetV2 backbone and our lightweight decoder); No.2. Baseline + NEEM; No.3. Baseline + GSIB; No.4. Baseline + NEEM + GSIB (our final model). The quantitative results and computational complexity on the Uneven_Mix dataset are summarized in Table 4. As shown, each module contributes positively to the overall performance of the model in detecting Mura defects. The baseline model (No.1), which consists of a MobileNetV2 backbone and a lightweight decoder, achieves a reasonable balance between speed and accuracy. However, introducing NEEM (No.2) results in improved pixel-level performance, especially in the F_m^{mean} score (0.887) and object-level precision (0.818), confirming the effectiveness of NEEM in refining local defect boundaries. However, the GSIB-enhanced model (No.3) offers even greater improvements in both object-level (precision: 0.831, recall: 0.804) and pixel-level (F_m^{mean}: 0.908, recall: 0.611) performance, highlighting its capability to capture large-scale brightness inconsistencies. Finally, the integrated model (No.4), combining both NEEM and GSIB, achieves the best performance across almost all metrics, including the highest object-level precision (0.859) and recall (0.823), highest pixel-level precision (0.824) and F_m^{mean} (0.632), and the lowest mean absolute error (M = 0.042). These results confirm the complementary strengths of the two modules: GSIB enhances global context awareness, while NEEM improves fine-grained boundary localization. Their integration leads to a robust and accurate Mura defect detection framework, demonstrating the effectiveness of our modular design.

4.4 Feature visualization.

Figure 6 illustrates the comprehensive visual analysis of the proposed method across various Mura defect scenarios. From top to bottom, the rows correspond to the original input images, ground truth (GT) segmentation masks, predicted segmentation outputs (Prediction), attention activation maps from three decoders (De1_attn, De2_attn, De3_attn), ground truth edge maps (Edge_GT), and the corresponding predicted edges (Edge_Pre). The segmentation results demonstrate high consistency with the ground truth, indicating the model's strong capability in accurately localizing Mura defects across different background conditions and defect morphologies. The attention maps reveal a progressive refinement of spatial focus through the decoder hierarchy, with deeper stages (De2 and De3) exhibiting enhanced sensitivity to defect contours and intensity transitions. Furthermore, the predicted edge maps show a close correspondence to the ground truth boundaries, validating the efficacy of the proposed edge-guided refinement strategy in enhancing boundary localization accuracy. Collectively, these visualizations substantiate the

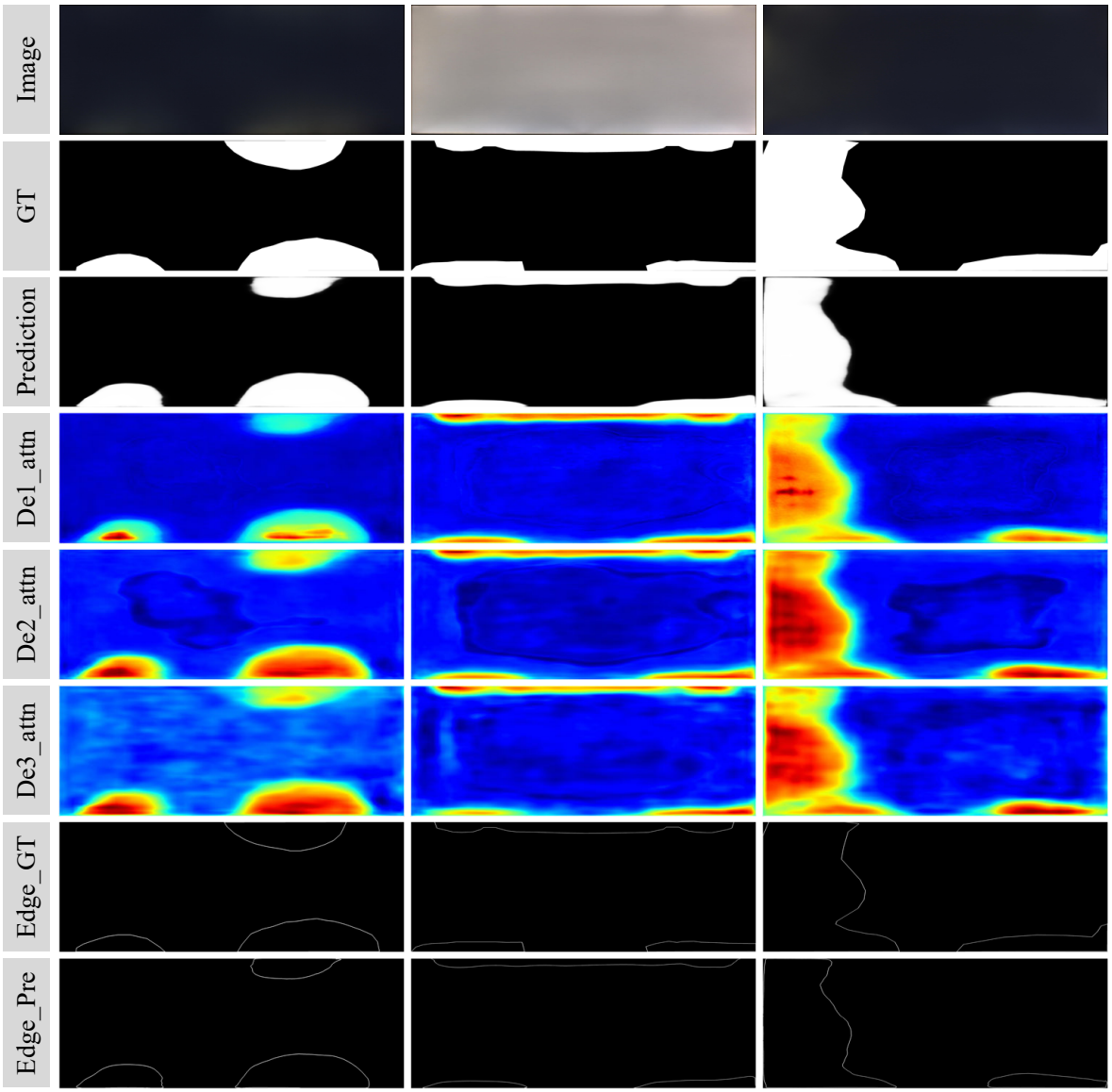

Figure 6. Visualization of segmentation, attention, and edge prediction across Mura defect types, demonstrating the effectiveness of the proposed multi-stage fusion and edge-aware architecture.

effectiveness of our attention-enhanced multi-stage decoder and edge-aware learning framework in achieving fine-grained, structurally coherent Mura defect segmentation.

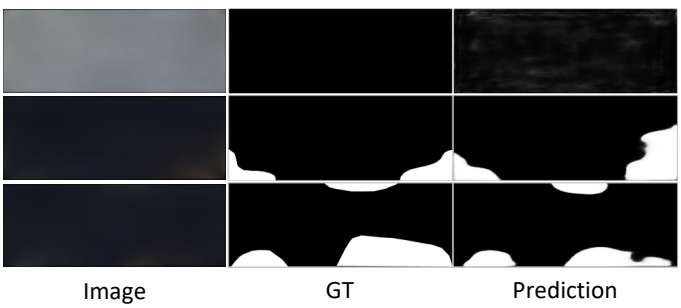

Figure 7. Some failure cases.

4.5 Failure Case

Although our model outperforms state-of-the-art (SOTA) methods in both qualitative and quantitative evaluations, several cases where detection results are less satisfactory are illustrated in Figure 7. Specifically, the first row presents a smooth image without visible defects, consistent with the ground truth. However, the prediction map exhibits some uncertain responses, suggesting that the model may be less reliable in defect-free white screen scenarios. This limitation could potentially be addressed by expanding the annotated white screen dataset. In the second row, slight defects are present and correctly labeled in the ground truth. The model detects part of the defective area but does not achieve full coverage. The third row shows clear defects, which

the model identifies well, closely matching the ground truth, although minor omissions or over-detections remain. Enhancing image contrast may help highlight Mura regions and further improve detection accuracy. Moreover, Table 5 compares the performance of SIFNet on the original Uneven_Black dataset versus the augmented Black dataset (Black_aug; 1,500 images), reporting improvements across all evaluated metrics: Precision (P), Recall (R), S-measure [35] (S_m), mean F-measure [36] (F_β^{mean}), mean E-measure [37] (E_ϕ^{mean}), max F-measure (F_β^{max}), max E-measure (E_ϕ^{max}), adaptive F-measure (F_β^{adp}), adaptive E-measure (E_ϕ^{adp}), and mean absolute error [38] (M). The results demonstrate SIFNet's consistent gains on the augmented dataset while preserving computational efficiency for industrial applications.

5 Conclusion

This study proposes SIFNet, an efficient deep learning framework for detecting large-area Mura defects in industrial applications. By combining a MobileNet-V2 backbone with our novel GSIB and NEEM modules, SIFNet achieves accurate defect localization while maintaining low computational costs (3.92M parameters, 6.89 GFLOPs). Experimental results demonstrate its effectiveness in handling Mura defects with smooth gradients and blurred boundaries, making it suitable for industrial deployment. The framework provides a practical solution for quality inspection while balancing accuracy and efficiency.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported by the YangFan Project of Guangdong Province of China under Grant 2020-05.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Li, J., Yin, Y., & Meng, H. (2024). Research progress of color photoresists for TFT-LCD. *Dyes and Pigments, 225*, 112094. [CrossRef]

[2] Cen, Y.-G., Zhao, R.-Z., Cen, L.-H., Cui, L.-H., Miao, Z.-J., & Wei, Z. (2015). Defect inspection for TFT-LCD images based on the low-rank matrix reconstruction. *Neurocomputing, 149*, 1206–1215. [CrossRef]

[3] CHEN, Z., SHEN, Y., ZHAI, C., DONG, C., & WANG, S. (2024). Development of AOI inspection of Mura defects on TFT-LCD surface. *Chinese Journal of Liquid Crystals and Displays, 39*(11), 1463-1476. [CrossRef]

[4] Xia, Y., Luo, C., Zhou, Y., & Jia, L. (2022). A hybrid method of frequency and spatial domain techniques for TFT-LCD circuits defect detection. *IEEE Transactions on Semiconductor Manufacturing*, 36(1), 45–55. [CrossRef]

[5] Ming, W., Zhang, S., Liu, X., Liu, K., Yuan, J., Xie, Z., ... & Guo, X. (2021). Survey of mura defect detection in liquid crystal displays based on machine vision. *Crystals, 11*(12), 1444. [CrossRef]

[6] Lee, J. Y., & Yoo, S. I. (2004). Automatic detection of region-mura defect in TFT-LCD. *IEICE TRANSACTIONS on Information and Systems, 87*(10), 2371-2378.

[7] Lin, Z. H., Lai, Q. Y., & Li, H. Y. (2024). A machine-learning strategy to detect Mura defects in a low-contrast image by piecewise gamma correction. *Sensors, 24*(5), 1484. [CrossRef]

[8] Chen, C., Chen, K. S., Vechet, S., & Guo, Y. J. (2024, December). An Integrated LCD Quality Assessment Flow based on the Integration of Image Processing, Ergonomics, and Machine Learning. In *2024 21st International Conference on Mechatronics-Mechatronika (ME)* (pp. 1-6). IEEE. [CrossRef]

[9] Luo, S., Chen, H., & Liu, B. (2025). DACA-Net: Detail-aware network with contrast attention for locating liquid crystal display defects. *Displays, 87*, 102913. [CrossRef]

[10] Jocher, G., Stoken, A., Borovec, J., Changyu, L., Hogan, A., Diaconu, L., ... & Dave, P. (2020). ultralytics/yolov5: v3. 0. *Zenodo*. [CrossRef]

[11] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(2), 318-327. [CrossRef]

[12] Yang, H., Mei, S., Song, K., Tao, B., & Yin, Z. (2017). Transfer-learning-based online Mura defect classification. *IEEE Transactions on Semiconductor Manufacturing, 31*(1), 116-123. [CrossRef]

[13] Lu, H.-P., & Su, C.-T. (2021). CNNs combined with a conditional GAN for Mura defect classification in TFT-LCDs. *IEEE Transactions on Semiconductor Manufacturing, 34*(1), 25–33. [CrossRef]

[14] Chen, P., Chen, M., Wang, S., Song, Y., Cui, Y., Chen, Z., Zhang, Y., Chen, S., & Mo, X. (2024). Real-time defect detection of TFT-LCD displays using a lightweight network architecture. *Journal of Intelligent Manufacturing, 35*(3), 1337–1352. [CrossRef]

[15] Khanam, R., & Hussain, M. (2024). Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*.

[16] Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-time flying object detection with YOLOv8. *arXiv preprint arXiv:2305.09972*.

[17] Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., & Chen, L. (1801). Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. CoRR abs/1801.04381 (2018). *arXiv preprint arXiv:1801.04381*.

[18] Sun, Y., Li, X., & Xiao, J. (2019). A cascaded Mura defect detection method based on mean shift and level set algorithm for active-matrix OLED display panel. *Journal of the Society for Information Display, 27*(1), 13-20. [CrossRef]

[19] Chen, L. C., & Kuo, C. C. (2007). Automatic TFT-LCD mura defect inspection using discrete cosine transform-based background filtering and 'just noticeable difference'quantification strategies. *Measurement Science and Technology, 19*(1), 015507. [CrossRef]

[20] Tsai, D. M., Chuang, S. T., & Tseng, Y. H. (2007). One-dimensional-based automatic defect inspection of multiple patterned TFT-LCD panels using Fourier image reconstruction. *International journal of production research, 45*(6), 1297-1321. [CrossRef]

[21] Cui, Y., Wang, S., Wu, H., Xiong, B., & Pan, Y. (2021). Liquid crystal display defects in multiple backgrounds with visual real-time detection. *Journal of the Society for Information Display, 29*(7), 547-560. [CrossRef]

[22] Chen, M., Chen, P., Wang, S., Cui, Y., Zhang, Y., & Chen, S. (2022). TFT-LCD mura defect visual inspection method in multiple backgrounds. *Journal of the Society for Information Display, 30*(11), 818-831. [CrossRef]

[23] Zhu, H., Huang, J., Liu, H., Zhou, Q., Zhu, J., & Li, B. (2021). Deep-learning-enabled automatic optical inspection for module-level defects in LCD. *IEEE Internet of Things Journal, 9*(2), 1122-1135. [CrossRef]

[24] Çelik, A., Küçükmanisa, A., Sümer, A., Çelebi, A. T., & Urhan, O. (2022). A real-time defective pixel detection system for LCDs using deep learning based object detectors. *Journal of Intelligent Manufacturing, 33*(4), 985-994. [CrossRef]

[25] Lin, G., Kong, L., Liu, T., Qiu, L., & Chen, X. (2022). An antagonistic training algorithm for TFT-LCD module mura defect detection. *Signal Processing: Image Communication, 107*, 116791. [CrossRef]

[26] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023, June). YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7464-7475). IEEE. [CrossRef]

[27] Wang, C. Y., Yeh, I. H., & Mark Liao, H. Y. (2024, September). Yolov9: Learning what you want to learn using programmable gradient information. In *European conference on computer vision* (pp. 1-21). Cham: Springer Nature Switzerland. [CrossRef]

[28] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., & Han, J. (2024). Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems, 37*, 107984-108011.

[29] Cheng, B., Liu, Z., Wang, Q., Shen, T., Fu, C., & Tian, A. (2024). Lightweight Progressive Multilevel Feature Collaborative Network for Remote Sensing Image Salient Object Detection. *IEEE Transactions on Geoscience and Remote Sensing, 62*, 1–17. [CrossRef]

[30] Liu, J., He, J., Chen, H., Yang, R., & Huang, Y. (2025). A Lightweight Semantic-and Graph-Guided Network for Advanced Optical Remote Sensing Image Salient Object Detection. *Remote Sensing, 17*(5), 861. [CrossRef]

[31] Feng, M., Lu, H., & Ding, E. (2019). Attentive Feedback Network for Boundary-Aware Salient Object Detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1623–1632). [CrossRef]

[32] Canny, J. (2009). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698. [CrossRef]

[33] Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., & Jagersand, M. (2019). BASNet: Boundary-Aware Salient Object Detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7471–7481). [CrossRef]

[34] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems, 25*.

[35] Fan, D. P., Cheng, M. M., Liu, Y., Li, T., & Borji, A. (2017, October). Structure-Measure: A New Way to Evaluate Foreground Maps. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 4558-4567). IEEE. [CrossRef]

[36] Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009, June). Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 1597-1604). IEEE. [CrossRef]

[37] Fan, D. P., Gong, C., Cao, Y., Ren, B., Cheng, M. M., & Borji, A. (2018, July). Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 698-704). [CrossRef]

[38] Perazzi, F., Krähenbühl, P., Pritch, Y., & Hornung, A. (2012). Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 733–740). [CrossRef]

[39] He, J., Liu, B., & Chen, H. (2025). HDPNet: Hourglass Vision Transformer with Dual-Path Feature Pyramid for Camouflaged Object Detection. In *2025 IEEE/CVF*

Winter Conference on Applications of Computer Vision (*WACV*) (pp. 8638–8647). [CrossRef]

[40] Tang, Z., Gao, Y., Zhu, Y., Zhang, Z., Li, M., & Metaxas, D. (2021, October). CrossNorm and SelfNorm for Generalization under Distribution Shifts. In *2021 IEEE/CVF International Conference on Computer Vision* (*ICCV*) (pp. 52-61). IEEE. [CrossRef]

Zhixi Wang is currently a PhD candidate with the School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu, China. His main research interests include display screen production and development, and intelligent manufacturing. (Email: Wangzx.rd@trulyopto.cn)

Jinpeng He is currently an M.S. candidate in Information and Communication Engineering at University of Electronic Science and Technology of China, Chengdu, China. (Email: jphe@std.uestc.edu.cn)

Huaixin Chen received the Phd. degree in optics engineering from University of Sichuan,China, in 2001, is Professor of university of electronic science and technology of China. (Email: huaixinchen@uestc.edu.cn)