Fine-Grained Hierarchical Perception Siamese Network for Battery Surface Defect Detection

***Abstract*—Battery surface defect detection is an essential part of maintaining quality standards in battery production. However, challenges persist when addressing defects with complex and variable shapes, particularly when there is low separability between the defects and the background. To tackle these issues, this paper introduces the Fine-Grained Hierarchical Perception Siamese Network for Battery Surface Defect Detection (FHS-Net), which detects defects by leveraging the differences between the inspected image and a reference image. The framework comprises a weight-sharing visual encoder, a Fine-Grained Enhancement Module (FGEM), and a Hierarchical Perception Decoder (HPD). Specifically, FGEM fuses adjacent stage feature maps from the backbone to capture fine-grained details at shallow levels and semantics at deeper levels, improving defect feature extraction. HPD calculates loss at each decoding layer, forming a hierarchical perception that enhances defect awareness from intermediate layers, progressively improving recognition accuracy. Furthermore, to enrich defect detection datasets, we collected a battery surface defect dataset named BSD, which consists of 2,183 images obtained from actual production lines using an automated optical inspection (AOI) camera. Comprehensive experiments performed on the BSD, NEU-SEG, and Magnetic-Tile-Defect datasets highlight the outstanding performance of the proposed framework.**

***Index Terms*—Battery surface defect detection, siamese network, fine-grained enhancement, hierarchical perception.**

I. Introduction

SURFACE defects in batteries diminish their lifespan and elevate the potential risk of explosion. Therefore, detecting surface defects on batteries is a crucial step in quality control during the manufacturing process [1]. Early battery surface defect detection systems relied on cameras for image acquisition and manual inspection, which had several drawbacks, including strong subjectivity, high labor intensity, and low efficiency [2]. Consequently, the development of an automated battery defect detection system that does not require human intervention is highly valuable. Although traditional machine learning methods [3]–[5] offer advantages such as computational simplicity, ease of implementation, and high interpretability, they also have limitations, including the need for complex handcrafted feature extraction, a high dependency on preprocessing, and poor robustness [6].

Deep learning [7] methods have seen substantial progress in the area of defect detection recently. Such methods can be classified into four primary strategies: classification [8]–[12], object detection [13], [14], segmentation [15]–[19], and anomaly detection [20]–[23]. Classification networks can classify images directly but are unable to pinpoint the exact location of defects. Object detection algorithms can not only classify images but also locate defects, however, they do not provide detailed contour information. Anomaly detection algorithms are capable of identifying unknown defect types and can detect defect contours, but when there is significant product variation, these algorithms may yield high false-positive rates. Segmentation-based defect detection algorithms effectively identify and accurately delineate defect contours by assigning semantic labels to each pixel.

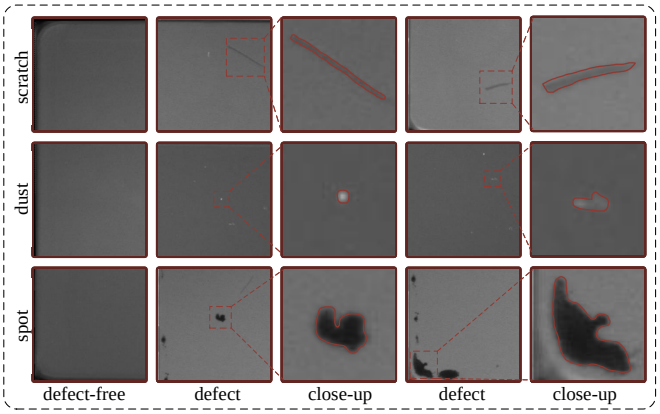

Fig. 1. Samples from the proposed battery surface defect detection dataset. The first column displays the normal dataset, for each row, one defect-free image is shown alongside two defective images, with the defect regions enlarged and annotated with precise pixel-level labels. This dataset presents challenges with complex and variable defect sample shapes, as well as the low separability between the defects and the background.

Convolutional neural network (CNN)-based segmentation networks commonly adopt an encoder-decoder architecture [15] and have shown promise in defect detection tasks. However, these approaches typically attempt to implicitly memorize defect patterns by learning intrinsic morphological features from extensive datasets. This learning strategy often fails to accurately detect defects with low separability or low saliency, as the network struggles to extract discriminative features solely from the target image when the defect shares similar visual attributes with the background. Given the scarcity of publicly available datasets for battery defect detection, we built the Battery Surface Defect Detection (BSD) dataset. The images were captured using an automated optical inspection (AOI) camera on actual production lines. Sample data are illustrated in Fig. 1. The BSD dataset presents several challenges: 1) defect samples exhibit complex and variable shapes; 2) there is low separability between the defects and the background. The types of defects included in the images primarily consist of dust, scratches, and spots.

To address these challenges, we design a Siamese network-based framework that analyzes a target image and a reference image by examining their differences, enabling the identification of defects that are otherwise indistinguishable from the

The code and datasets will be released upon acceptance.

background. In production settings, product designs are typically based on strictly standardized normal templates, making it practical and straightforward to select a defect-free sample as a reference. Specifically, we propose the Fine-Grained Hierarchical Perception Siamese Network (FHS-Net), inspired by the Siamese network architecture [24], which leverages normal samples as references and processes both test and reference images to accurately segment defects, particularly those that are small in size or exhibit low separability from the background. The primary contributions of this work are summarized as follows:

1) A novel framework, FHS-Net, is introduced for defect detection by analyzing discrepancies between a target image and a reference image, consisting of a shared-weight backbone, a Fine-Grained Enhancement Module (FGEM), and a Hierarchical Perception Decoder (HPD).
2) A Fine-Grained Enhancement Module (FGEM) is proposed to address low separability and complex defect shapes by integrating features from adjacent stages, enabling the capture of fine-grained details and deep semantic information.
3) A Hierarchical Perception Decoder (HPD) is designed to progressively enhance detection accuracy by calculating loss at each decoding layer, establishing a hierarchical perception mechanism that facilitates intermediate defect detection and early weak signal capture.
4) A dedicated Battery Surface Defect (BSD) dataset is constructed, containing 2,183 images, to support binary segmentation tasks. Extensive experiments demonstrate that the proposed method achieves competitive performance on BSD, NEU-SEG, and Magnetic-Tile-Defect datasets.

The subsequent sections of this paper are arranged as follows. Section II reviews four categories of deep learning-based industrial defect detection approaches, including classification, object detection, segmentation, and anomaly detection methods. Section III describes the battery surface defect dataset. Section IV provides a description of our framework. Section V presents the experimental results. Section VI presents the limitations and future work. Section VII serves as the conclusion of the paper.

II. RELATED WORKS

In this section, we review industrial defect detection methods, which can be broadly categorized into classification-based, object-detection-based, anomaly-detection-based, and segmentation-based methods.

A. Classification-Based Industrial Defect Detection

In the early stages of industrial defect inspection, supervised classification models played a foundational role. Pioneering CNN, such as VGG [25], demonstrated the transformative potential of deep learning in visual recognition, leading to their widespread application in quality control within manufacturing processes. ResNet [26], with its innovative residual learning framework, further enhanced the trainability of deep architectures and has since become a cornerstone for defect classification tasks. Meng et al. [9] introduced a method that leverages deep residual convolutions to automatically and accurately assess the severity of metal surface flaws. Fu et al. [10] proposed a strategy that emphasizes the learning of low-level features combined with multi-receptive field modules to mitigate challenges such as sensitivity to illumination variations, noise interference, and limited real-time detection capabilities. With the increasing concern over data scarcity, recent studies have shifted toward self-supervised and few-shot learning techniques for defect detection [11], [12]. Nevertheless, although classification-based networks can determine the presence of defects, they often struggle to pinpoint the precise location of these anomalies within an image.

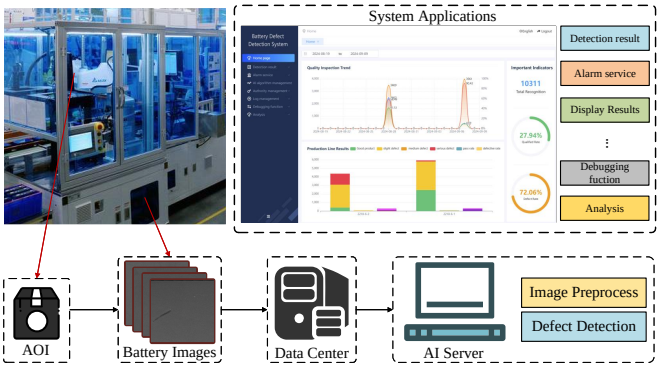

Fig. 2. Battery surface defect inspection system. The system first captures images of the battery surface and transmits the data to the data center for storage. Next, the AI server automatically conducts data preprocessing and defect detection. Finally, the detection results are visualized through an intuitive front-end interface, allowing management personnel to analyze the data and make informed decisions.

B. Object-Detection-Based Industrial Defect Detection

In contrast to classification-based techniques, object detection approaches provide the ability to localize defects within images, effectively addressing the limitations of classifiers that lack spatial detail. Object detection algorithms are broadly categorized into two paradigms: two-stage detectors, such as Faster R-CNN [27], and one-stage detectors, represented by the YOLO family of methods [28], [29]. Due to their high inference efficiency and lightweight deployment characteristics, YOLO-based frameworks have become the predominant choice in industrial defect detection scenarios. Numerous researchers have sought to enhance detection accuracy by refining the YOLO architecture. For instance, Zhong et al. [13] incorporated a coordinate attention mechanism and a feature channel shuffle operation into an improved variant of YOLOv8m, aiming to overcome challenges in insulator defect detection, including the trade-off between real-time performance and accuracy, data scarcity, and the influence of varying weather conditions. Similarly, Bao et al. [14] proposed an enhancement to YOLOv4 based on a channel-and-coordinate-aware attention strategy, which effectively addressed issues in photovoltaic cell inspection, such as poor multi-scale defect recognition, suboptimal feature representation, and insufficient model generalization. Despite these advancements, object detection models primarily yield bounding boxes and are

therefore limited in their ability to provide precise contour-level delineation of defect shapes.

C. Anomaly-Detection-Based Industrial Defect Detection

Anomaly detection techniques are designed to learn representations solely from normal samples and identify abnormal regions by detecting deviations from typical patterns. This approach effectively addresses the challenge of limited defect data and facilitates both image-level and pixel-level AUROC evaluation. By establishing appropriate thresholds, these methods can perform both defect classification and segmentation tasks [20]. Current anomaly detection approaches can be broadly categorized into three paradigms: reconstruction-based methods, normalizing flow-based frameworks, and knowledge distillation-based strategies. For instance, Zavrtanik et al. [21] Proposed a reconstruction-driven anomaly detection method that frames the task as an image inpainting problem. Their technique involves randomly masking regions of an image and restoring them, with anomalies inferred by analyzing discrepancies between the original and reconstructed content. Zhou et al. [20] introduced a multi-scale normalizing flow framework that extracts hierarchical feature representations and processes them through asymmetric parallel and fusion pathways. This cross-scale interaction significantly enhances the ability to detect and localize anomalies of varying sizes. Furthermore, Zhu et al. [22] developed an asymmetric teacher-student feature pyramid matching network to address the limitations of conventional distillation-based methods. Their approach enhances anomaly representation by mitigating the drawbacks of symmetric architectures, including indistinct feature differences, poor precision in deeper layers, and incomplete knowledge transfer resulting from similarities in shallow features. Although anomaly detection methods can identify previously unseen defect types and delineate their contours, their performance often relies on manually defined thresholds. As a result, under conditions of significant variation in product appearance, these methods may experience an increased false positive rate.

D. Segmentation-Based Industrial Defect Detection

In contrast to these methods, industries prefer segmentation algorithms, as defect detection based on segmentation networks can effectively identify and accurately delineate defect contours. In recent years, segmentation-based defect detection algorithms have made significant advancements. For instance, Zou et al. [15] developed the DeepCrack network, which is based on SegNet and integrates features from both encoders and decoders to effectively capture subtle crack structures in roads. Nevertheless, strictly tailored to thin topological structures, it often fails to segment complex, variable-shaped defects indistinguishable from the background. While Dong et al. [16] proposed a pyramid feature fusion and global context attention network to alleviate detection complexity, their excessive reliance on global context tends to suppress high-frequency local features, resulting in the loss of fine-grained edge details in subtle defects. Zhong et al. [17] presented a Gaussian attention network based on DeepLabv3+,

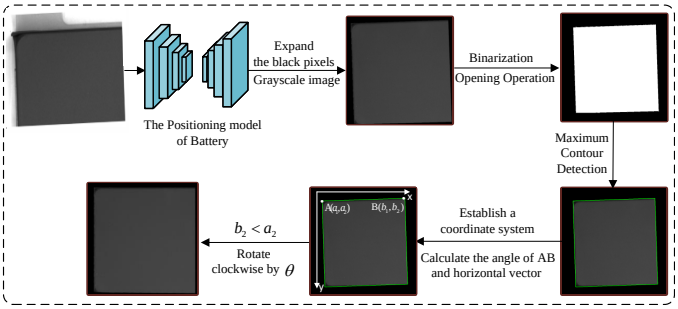

Fig. 3. Data Preprocessing Steps.

Algorithm 1 Image alignment

1: **Input:** RGB image I
2: **Output:** Corrected image I_{corr}
3: **Step 1:** Expand the black pixels and convert to grayscale:
$$I_{\text{gray}} = \text{RGB2Gray}(\text{Expand}(I))$$
4: **Step 2:** Binarize the grayscale image and morphologically open to remove noise:
$$I_{\text{denoised}} = \text{MorphOpening}(\text{Binarize}(I_{\text{gray}}))$$
5: **Step 3:** Extract the largest contour:
$$\text{contours} = \text{FindContours}(I_{\text{denoised}})$$
$$\text{largest\_contour} = \text{MaxContour}(\text{contours})$$
6: **Step 4:** Compute the minimum bounding rectangle for the largest contour, extract the coordinates of the top-left corner A(a_1, a_2) and the top-right corner B(b_1, b_2), and calculate the angle θ between vector $\overrightarrow{AB}$ and the horizontal vector $\overrightarrow{e}$:
$$\theta = \arccos \frac{\overrightarrow{AB} \cdot \vec{e}}{\left\|\overrightarrow{AB}\right\| \cdot \|\vec{e}\|}$$
7: **Step 5:** Calculate rotation matrix R and transform the image using an affine transformation:
$$I_{\text{corr}} = \text{AffineTransform}(I, R)$$
8: **Return:** I_{corr}

which captures long-range dependencies for global information extraction, effectively addressing issues related to fuzzy defect areas and irregular sizes and shapes in wood surface defect segmentation. While effective globally, the Gaussian kernels risk oversmoothing feature maps, failing to preserve sharp boundaries required for low-contrast defect detection. Additionally, Zhu et al. [30] proposed the DS^{2}A-Former, a dual-stream spatial attention Transformer network that utilizes a Dual-Scale Spatial Cross-Attention module to integrate semantic and spatial information for battery defect segmentation. However, the inherent patch-based tokenization of Transformer architectures often disrupts local pixel continuity, hindering the precise recovery of fine-grained edge details necessary for high-precision metrology. Recently, some researchers have explored defect detection networks based on template-based Siamese networks. Chen et al. [31] proposed the Background Generalization Network, a Siamese-based framework for surface defect segmentation in SMD chips. It integrates self- and cross-attention mechanisms for dense feature extraction,

and employs dual-softmax with Mutual Nearest Neighbour matching to suppress background noise. Despite its efficacy in background suppression, this paradigm overlooks intrinsic feature perception in intermediate layers, limiting the capture of details in morphologically variable defects.

Battery defect samples exhibit complex, variable shapes and low separability with the background, while prior approaches have overlooked the fine-grained edge details of these defects. To overcome the aforementioned challenges, we propose a FGEM designed to capture fine-grained details and deep semantic information. Additionally, we introduce a HPD to enhance defect perception capabilities within the intermediate layers of the network.

III. BATTERY SURFACE DEFECT DATASET

This section provides a detailed description of the preprocessing process of the collected dataset, followed by a detailed explanation of the processed dataset.

A. Images Preprocessing

We have developed a battery surface defect detection system, and Fig. 2 illustrates the detailed architecture of the system. To mitigate background interference and address the extreme class imbalance between defect and background pixels, a Region-of-Interest extraction pipeline is implemented. A lightweight segmentation network isolates the battery surface from the complex production environment, ensuring the subsequent defect detection model focuses exclusively on relevant texture features. Due to inaccuracies in mechanical positioning, conveyor belt deviations, and operator errors, AOI-captured images may exhibit angular misalignments [32]. Therefore, image alignment is a crucial preprocessing step to enhance defect detection performance. The steps for image alignment, as shown in Algorithm 1.

The flowchart of the above process is shown in Fig. 3. During production inference, test images first undergo localization and alignment preprocessing before being fed into the proposed framework. The alignment algorithm is adaptable to other shape-regularized industrial products, such as PCBs, by identifying the coordinates of parallel edges (e.g., points A and B). While morphological methods are optional and often require handcrafted feature extraction and threshold tuning, keypoint detection [33] can also be used to locate these points, albeit at the cost of extensive data annotation. In practice, excessive reliance on deep models may impact system efficiency. Thus, method selection should be tailored to the specific constraints of the production line.

B. The details of BSD Dataset

Given the scarcity of publicly available datasets for battery defect detection, we built the Battery Surface Defect Detection (BSD) dataset. After rectifying the images, pixel-level annotation was performed using Labelme. Specifically, three main types of defects are clearly identifiable: spot, dust and scratches. Some samples from the BSD dataset are presented in Fig. 1. The dataset poses challenges due to complex defect shapes and low background separability. Since ambiguous visual characteristics make sub-type classification difficult, and industrial QC prioritizes anomaly detection over categorization, we formulate the task as binary segmentation. The dataset contains 229 original images, each with a resolution of 1169×847 pixels, captured from different battery units to ensure sample diversity. To ensure the rigor of the evaluation and prevent potential data leakage caused by cropping, we strictly partitioned the dataset at the original image level. Specifically, the 229 original images were randomly divided into training, validation, and test sets with a ratio of 6:2:2, resulting in 137, 46, and 46 images, respectively. These images were manually selected from over 50,000 inspection images collected on the production line, with each selected battery unit exhibiting at least one visible surface defect. The defect-free reference image was captured using the same AOI equipment, lighting conditions, and camera angles as those used for the inspected samples, and was manually verified to be free of visible anomalies in accordance with the quality control (QC) standards adopted in battery production. Due to the stringent manufacturing controls that ensure visual uniformity among qualified batteries, a single, QC-verified image is deemed representative of the defect-free class. Given the large size of the original images, we employed a sliding-window cropping strategy with overlapping. Specifically, we set the cropping stride to be smaller than the patch size (256×256) to create overlapping regions between adjacent patches. This design guarantees that any defect located at the edge of one patch is captured near the center of an adjacent overlapping patch, preserving the completeness of defect features. As a result, we constructed the final dataset consisting of 2,183 sub-images. The dataset's final distribution is shown in Table I, and the BSD dataset distribution is visualized in Fig. 5. Overall, the availability of this dataset is anticipated to significantly contribute to the advancement of research in surface defect segmentation technology.

IV. PROPOSED METHOD

This section offers a brief overview of the data flow in the proposed FHS-Net architecture, along with an in-depth analysis of each component of the model.

A. Overall Architecture of FHS-Net

To detect low-separability and small defects, we propose a novel framework, FHS-Net, to analyze both the target and a reference image, as illustrated in Fig. 11. FHS-Net takes both a target image and a reference image, utilizing the differences between them to identify defects. In this study, defect detection is framed as a binary segmentation task, and a lightweight backbone, MobileNetV2, is employed as the feature extractor. Given a target image and a reference image, the feature extractor consists of five stages, with the output feature map at each stage denoted as:

$$\{f_i^d\}, \{f_i^r\} = \mathrm{E}(I_d), \mathrm{E}(I_r) \quad i \in \{1, 2, 3, 4, 5\} \tag{1}$$

Where $\mathrm{E}(\cdot)$ represents the backbone, while and f_i^d and f_i^r denote the output feature maps at the i^{th} stage after processing

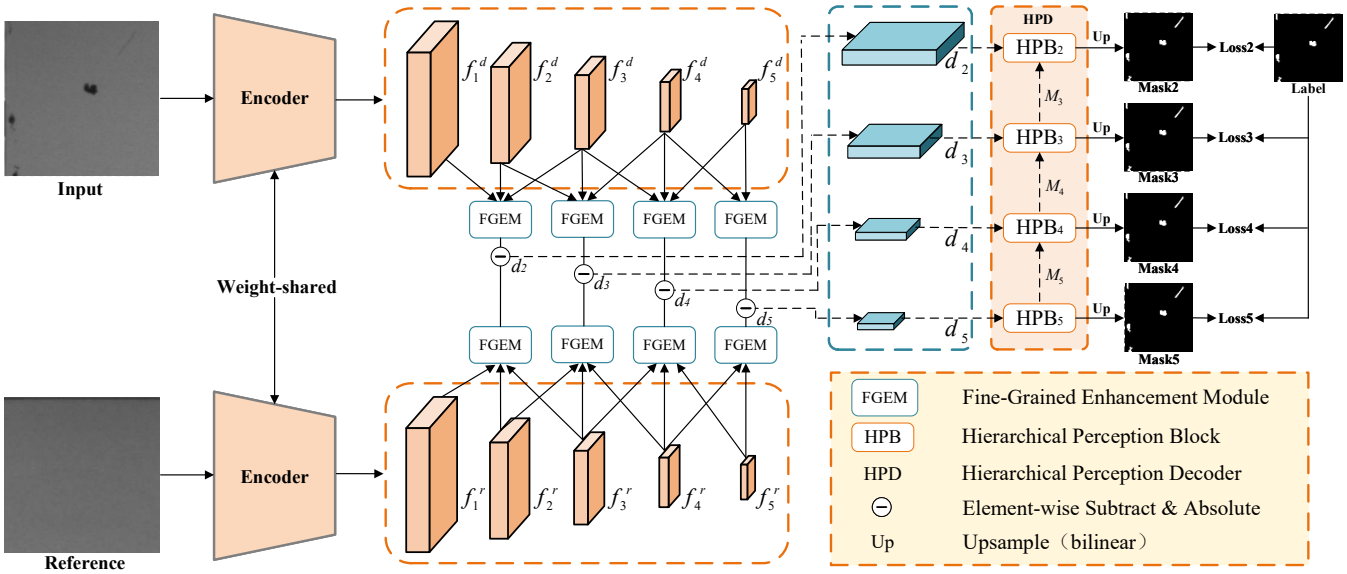

Fig. 4. Overview of the basic architecture of our proposed FHS-Net.

the target image and the reference image, respectively. The output feature map size at the i^{th} stage is $(256/2^i, 256/2^i)$, with output feature dimensions of 16, 24, 32, 96, and 320. Adjacent feature maps will be input into the FGEM for multi-layer feature extraction. The feature maps produced will merge spatial information from shallow features with the deep semantic information extracted from deeper layers. The newly obtained feature maps are represented as follows:

$$\hat{f}_i^k = \begin{cases} \text{FGEM}(f_{i-1}^k, f_i^k, f_{i+1}^k) & i \in \{2,3,4\} \\ \text{FGEM}(f_{i-1}^k, f_i^k) & i \in \{5\} \end{cases} \tag{2}$$

Where $\text{FGEM}(\cdot)$ refers to the Fine-Grained Enhancement Module Function, $\hat{f}_i^k$ denotes the output of FGEM, $k \in \{d, r\}$. This module effectively integrates deep and shallow feature maps, utilizing contextual information to capture low-separability and small defects. A comprehensive discussion of the FGEM will be provided in IV-B.

Instead of using $\hat{f}_i^d$ and $\hat{f}_i^r$ pairs for splicing, summing, multiplying, convolution fusion, and other operations, we leverage the reference image by performing absolute value subtraction on both images. This approach highlights the differences between the target image and the reference image, thereby facilitating the detection of minor defects. The resulting difference feature map is denoted as:

$$d_i = \mid \hat{f}_i^d - \hat{f}_i^r \mid \quad i \in \{2,3,4,5\} \tag{3}$$

Where $|\cdot|$ denotes the absolute value operation, and - represents element-wise subtraction.

During the backpropagation process, the gradients received by the network layers closer to the input diminish progressively. In deep learning models with long chains, this phenomenon, known as gradient vanishing, can impede convergence, and the residual structure effectively mitigates the gradient vanishing problem associated with long chains. Inspired by the above, our decoder comprises four Hierarchical

TABLE I
OVERVIEW OF BSD DATASET SPLIT AND STATISTICS

Split	Image Counts		Defect Instances		
	Normal	Defective	Dust	Scratch	Spot
Train	414	893	1558	116	318
Val	118	320	608	37	193
Test	163	275	516	34	79
Total	695	1488	2680	187	590

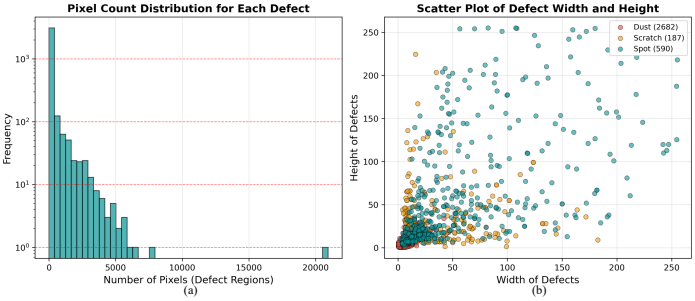

Fig. 5. The visualization of the BSD Dataset attributes. (a) The distribution of all defect instances. (b) The scatter plot of the width and height of the minimum bounding rectangles for all defect instances.

Perception Blocks (HPB) arranged in series. Each block, except for HPB_5, integrates the difference feature map from the current stage with the perceptual information from the preceding HPB. During pixel recovery, we compute loss for each HPB, enabling defect detection in intermediate layers, which mitigates the gradient vanishing problem and enhances recognition accuracy. This approach emphasizes fine adjustments in pixel recovery at each stage, ensuring that the model optimally aligns in the correct direction, allowing the intermediate layers to perceive potential defect areas. The

decoding phase of the model is represented as follows:

$$M_i = \begin{cases} \mathrm{HPB_i}(M_{i+1}, d_i) & i \in \{3,4\} \\ \mathrm{HPB_i}(d_i) & i \in \{5\} \end{cases} \tag{4}$$

Here, $\mathrm{HPB_i}(\cdot)$ denotes the function of the Hierarchical Perception Block, and M_i represents the output of the i^{th} HPB. IV-C will provide a detailed introduction to the Hierarchical Perception Decoder.

In the actual inference phase, we employed a sliding-window pipeline where the original full-resolution image is processed in overlapping patches. For the overlapping regions, we averaged the predicted probability maps from different patches instead of using simple binary overwriting.

B. Fine-Grained Enhancement Module

Battery surface defects are often small and exhibit diverse shapes, as shown in the second row of Fig. 1, where the pixel size of dust does not exceed 10 pixels. However, during the deepening of the network, continuous downsampling tends to weaken or even lose the fine-grained information of small defects, making accurate detection challenging. In the feature hierarchy, shallow layers preserve rich fine-grained details of defects, while deeper layers capture more abstract and semantic cues about defect locations [34]. Although FPN [35] facilitates multi-scale fusion via top-down pathways and lateral connections, its indirect cascaded propagation risks oversmoothing the subtle edge details essential for identifying tiny defects. Similarly, fusion strategies like ASFF [36] learn to spatially filter conflictive information across scales to resolve pyramidal inconsistencies, yet they prioritize global scale-invariance at the expense of the localized consistency required for capturing subtle defect boundaries. In contrast, FGEM fuses adjacent stage feature maps to integrate shallow details with deep semantics, thereby addressing the challenges of low separability and complex defect shapes by enhancing feature discriminability and preserving fine-grained boundary information. A detailed flowchart of FGEM is presented in Fig. 6.

To integrate features from adjacent stages $(f_{i-1}^k, f_i^k, f_{i+1}^k)$, we first standardize their spatial resolutions. Specifically, the shallow features f_{i-1}^k are downsampled via the Mp operation, while the deep features f_{i+1}^k are resized using the bilinear upsampling operator Up. A standard convolution block $\mathrm{C_{3\times3}}$ followed by a ReLU activation σ is then applied to unify the channel dimensions, yielding the aligned features ν:

$$\nu_{i-1}^k = \sigma(\mathrm{C_{3\times3}}(\mathrm{Mp}(f_{i-1}^k))) \quad i \in \{2,3,4,5\} \tag{5}$$
$$\nu_i^k = \sigma(\mathrm{C_{3\times3}}(f_i^k)) \quad i \in \{2,3,4,5\} \tag{6}$$
$$\nu_{i+1}^k = \mathrm{U}(\sigma(\mathrm{C_{3\times3}}(f_{i+1}^k))) \quad i \in \{2,3,4\} \tag{7}$$

Subsequently, these aligned features are concatenated to integrate multi-scale information. The resulting fusion is processed by convolution layers to generate the intermediate feature C_i^k

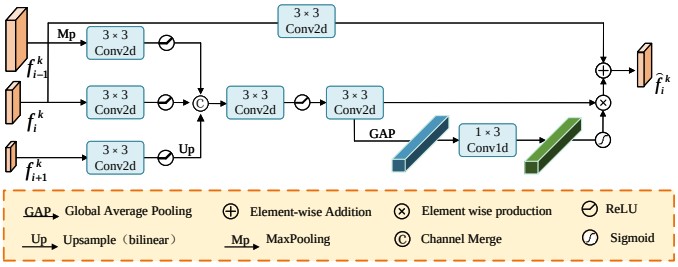

Fig. 6. Details of the proposed FGEM.

(Eq. 8-9). Note that for the final stage ($i = 5$), only the preceding and current features are concatenated:

$$\hat{\nu}_i^k = \begin{cases} \mathrm{Cat}(\nu_{i-1}^k, \nu_i^k, \nu_{i+1}^k) & i \in \{2,3,4\} \\ \mathrm{Cat}(\nu_{i-1}^k, \nu_i^k) & i \in \{5\} \end{cases} \tag{8}$$
$$C_i^k = \mathrm{C_{3\times3}}(\sigma(\mathrm{C_{3\times3}}(\hat{\nu}_i^k))) \quad i \in \{2,3,4,5\} \tag{9}$$

Finally, each feature dimension on the feature map does not provide the information we seek. Therefore, to emphasize discriminative channel features, we employ a channel attention mechanism followed by a residual connection. The attention map $atten$ recalibrates the feature responses, and the final enhanced feature $\hat{f}_i^k$ is obtained as:

$$atten = \mathrm{Sig}(\mathrm{C_{1\times3}}(\mathrm{G}(C_i^k)))$$
$$\hat{f}_i^k = C_i^k \otimes atten + \mathrm{C_{3\times3}}(f_i^k) \quad i \in \{2,3,4,5\} \tag{10}$$

Where $\mathrm{Cat}(\cdot)$ denotes channel concatenation, $\mathrm{G}(\cdot)$ is global average pooling, $\mathrm{Sig}(\cdot)$ is the Sigmoid function, and $\otimes$ denotes element-wise multiplication.

C. Hierarchical Perception Decoder

HPD establishes a hierarchical perception mechanism by enforcing supervision at every decoding stage. This architecture alters gradient propagation dynamics to counteract signal decay. Specifically, let $\mathcal{L}$ denote the objective function and w represent a weight in a shallow layer l. In a standard network supervised only by the final output, the gradient is governed by a long recursive product:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial o_N} \cdot \left(\prod_{k=l}^{N-1} \frac{\partial o_{k+1}}{\partial o_k} \right) \cdot \frac{\partial o_l}{\partial w} \tag{11}$$

As the network depth N increases, the continuous multiplication of partial derivatives, which are typically less than 1 due to activation saturation or weight initialization, leads to exponential gradient decay. In contrast, HPD imposes supervision at each HPB stage k. This strategy creates direct feedback routes, allowing gradients to backpropagate through shorter intermediate paths. Consequently, the total gradient becomes an additive aggregation of signals from all stages:

$$\frac{\partial \mathcal{L}}{\partial w} = \sum_{k=l}^{N} \frac{\partial \mathcal{L}_k}{\partial w} = \frac{\partial \mathcal{L}_l}{\partial w} + \sum_{k=l+1}^{N} \frac{\partial \mathcal{L}_k}{\partial w} \tag{12}$$

Crucially, the first term $\frac{\partial \mathcal{L}_l}{\partial w}$ functions as a "short-circuit" path that strictly bypasses the deep recursive chain, ensuring that shallow layers receive direct and strong supervision

signals. This explicit gradient preservation enables the model to perceive potential defect regions at intermediate layers. Unlike traditional deep supervision [37], which employs parallel auxiliary branches independent of the main decoder, HPD integrates supervision directly into a cascaded refinement chain. In this architecture, the prediction mask at each stage functions not only as a loss objective for optimization but also as an essential semantic prior that is fed back into the subsequent decoding block to guide feature reconstruction. Consequently, this design shifts the focus from mere gradient-level regularization to a structural recursive perception enhancement, enabling the model to refine its defect awareness progressively through the decoding hierarchy.

The decoder comprises four cascaded Hierarchical Perception Blocks (HPB), as illustrated in Fig. 7. Except for the fifth stage, the i-th HPB fuses the perception features M_{i+1} from the $(i+1)$-th block with the differential map d_i. The architecture processes these features through three parallel branches.

The rightmost branch generates the supervision mask and aligned features. A 1×1 convolutional layer $\mathrm{C}_{1\times1}$ compresses the channel dimension, followed by upsampling to produce the mask map Mask_i for loss calculation. Simultaneously, to ensure alignment with subsequent channels, the $\mathrm{C}_{3\times3}$ operation followed by ReLU is applied to generate feature E_i^1:

$$\mathrm{Mask}_i = \mathrm{Up}(\mathrm{C}_{1\times1}(M_{i+1} + d_i)) \tag{13}$$

$$E_i^1 = \sigma(\mathrm{C}_{3\times3}(\mathrm{C}_{1\times1}(M_{i+1} + d_i))) \tag{14}$$

The middle branch applies a direct convolutional refinement using the $\mathrm{C}_{3\times3}$ layer to obtain E_i^2. The leftmost branch employs a gating mechanism: the global average pooling operation G and the $\mathrm{C}_{1\times1}$ layer with Sigmoid activation produce channel weights, which are multiplied by M_{i+1} to yield E_i^3, emphasizing defect-related texture information:

$$E_i^2 = \sigma(\mathrm{C}_{3\times3}(M_{i+1} + d_i)) \tag{15}$$

$$E_i^3 = \mathrm{Sig}(\mathrm{G}(\mathrm{C}_{1\times1}(M_{i+1} + d_i))) \otimes M_{i+1} \tag{16}$$

Finally, the outputs from all branches are aggregated via summation:

$$M_i = E_i^1 + E_i^2 + E_i^3 \tag{17}$$

Specific adaptations are implemented at the boundaries. In the fifth stage, due to the absence of prior semantic features, d_5 is directly fed into the branches to generate Mask_5 and M_5. In the second stage, as M_2 is not required for further propagation, only the supervision mask is generated via the $\mathrm{C}_{3\times3}$ and $\mathrm{C}_{1\times1}$ layers followed by upsampling.

D. Loss Function

For battery surface defect detection tasks, the defect instances are typically very small and occupy a minimal area compared to the background, resulting in a significant class imbalance issue. To address this, we incorporate methods L_{ce} [38] and L_{dice} [39] to optimize the proposed framework. L_{ce} is a commonly used classification loss that measures the discrepancy between the predicted and true probability

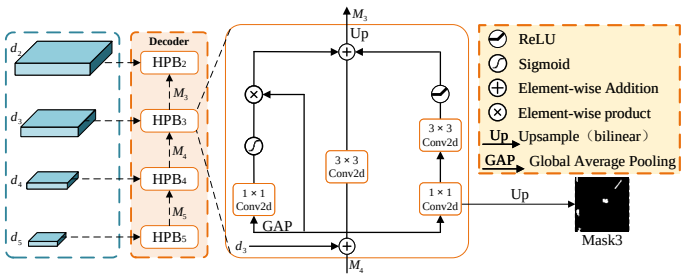

Fig. 7. Details of the proposed HPD.

distributions. The L_{dice} effectively balances the differences in sample quantities between categories and emphasizes the precise segmentation of small targets by focusing on the overlap of target regions. This process can be represented by Equations (18-20).

$$L_{ce}^i = -[G \cdot \log P_i + (1-G) \cdot \log(1-P_i)] \tag{18}$$

$$L_{dice}^i = 1 - 2 \cdot \frac{G \cdot P_i}{\|G\| + \|P_i\|} \tag{19}$$

$$L = \sum_{i=2}^{5} L_{ce}^i + \lambda L_{dice}^i \tag{20}$$

Here, G refers to the ground truth mask, where each element is represented by either 0 or 1, with 0 indicating the background and 1 indicating a defect. P_i represents the predicted mask at the i^{th} stage, where values closer to 1 indicate a higher defect probability. $\|\cdot\|$ denotes the L1 norm and λ is set to 1.

V. EXPERIMENTS

This section first introduces our experimental setup, three datasets for surface defect detection, and the evaluation metrics. We then perform ablation studies on the BSD dataset to evaluate the impact of the Fine-Grained Enhancement Module, Hierarchical Perception Decoder, and Siamese Structural components. We also study the effect of using different backbones. To assess the model's generalization ability, we compare its performance with that of 19 state-of-the-art algorithms across the three datasets.

A. Experimental Setup

For all experiments, a system equipped with an Intel Core i5-13400 CPU and an NVIDIA RTX 3060 12GB GPU was utilized. The software environment included PyTorch 1.13.1, CUDA 11.7, and Ubuntu 22.04. To train FHS-Net, the Adam optimizer was employed, running for a total of 30,000 iterations with an initial learning rate of 0.0005, weight decay of 0.0001, and momentum parameters set to (0.9, 0.99). A polynomial learning rate scheduling strategy was used, with a batch size set to 16. Ablation studies were conducted on the BSD dataset using FHS-Net, and comparative experiments were carried out on three datasets: BSD, NEU-SEG [40], and Magnetic-Tile-Defect [41]. BG-Net [31], FC-diff [42], and TFI-Net [43] were trained using the same configuration as FHS-Net. Other baseline methods were implemented using

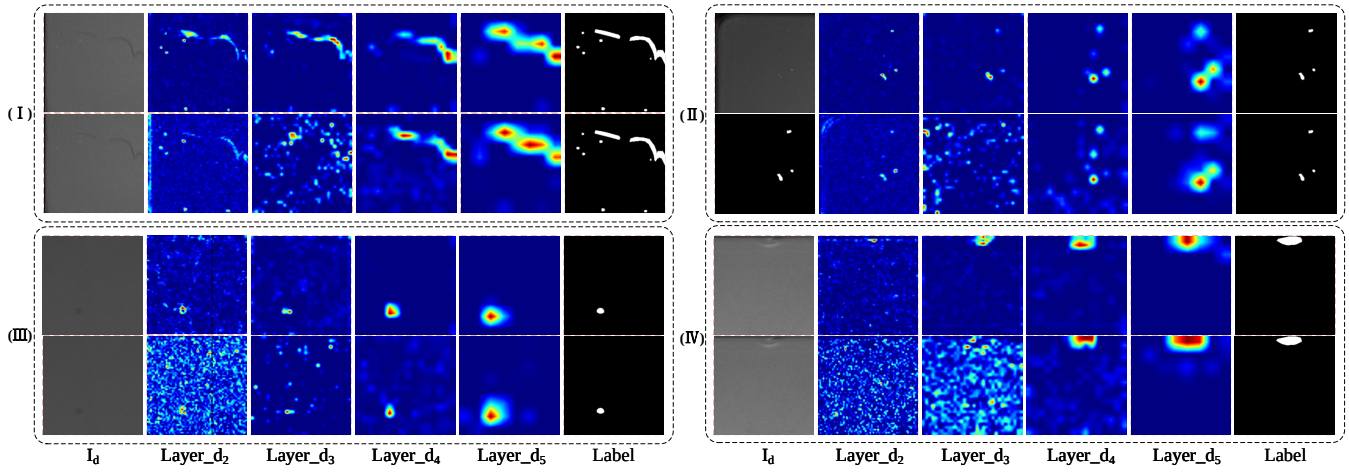

Fig. 8. Quantitative analysis results of the difference layers: Layer_d2, Layer_d3, Layer_d4, and Layer_d5 based on GradCam. Four sets of images (I), (II), (III), and (IV) are selected from the test set of BSD. The first row of each set represents FHS-Net, while the second row represents w/o_FGEM. The redder the image, the more attention the model pays to it. FGEM fuses the feature maps of neighboring phases to enhance the extracted fine-grained features while suppressing background interference factors.

the MMSegmentation framework, with preprocessing setting based on the ADE20K [44] configuration. For these methods, 12,000 iterations were performed with a batch size of 4. All models were evaluated using the checkpoint that achieved the best performance on the validation set, and tested on the official test split of each dataset.

B. Datasets

To evaluate the generalization and transferability of FHS-Net across different industrial scenarios, we conducted comparative experiments on three datasets: BSD, NEU-SEG [40], and Magnetic-Tile-Defect [41], all of which focus on fine-grained industrial surface defect detection. NEU-SEG comprises 4,470 grayscale images of surface defects in hot-rolled steel strips. All defect types were merged into a single class to simplify the binary segmentation task. A pseudo-reference was generated by modifying the smallest defect image using surrounding normal regions. The dataset was split into 2,790 training, 840 validation, and 840 test samples by extracting a validation subset from the original training set. The Magnetic-Tile-Defect dataset focuses on detecting surface defects in magnetic tile components. It contains 1,344 grayscale images. A defect-free image was randomly selected as the reference, and the dataset was divided into 805 training, 267 validation, and 272 test samples using a 6:2:2 ratio.

C. Evaluation Metrics

Performance: To assess the generalization ability of the model, we employed two metrics: the F1 score and IoU. The F1 score balances precision and recall, with higher values indicating better model accuracy in detecting true positives and minimizing false positives and negatives. The IoU, which measures the overlap between the predicted and ground truth regions, reflects the spatial accuracy of the segmentation. Higher IoU values indicate that the model is more accurately identifying and segmenting the relevant objects. For detailed definitions of the formulas, please refer to Equations (21-24).

$$Precision = \frac{TP}{TP + FP} \tag{21}$$

$$Recall = \frac{TP}{TP + FN} \tag{22}$$

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \tag{23}$$

$$IoU = \frac{TP}{TP + FP} \tag{24}$$

Where TP (True Positives) denotes the pixels correctly identified as defects, FP (False Positives) refers to pixels misclassified as defects, and FN (False Negatives) indicates the defect pixels that were not detected by the model.

Efficiency: While detection accuracy is vital, it is insufficient without considering real-time detection capabilities and hardware constraints. We evaluate the model's efficiency using four key metrics: Params (total trainable weights and biases), FLOPs (model's computational requirements), GPU Memory (memory consumption with a batch size of 1 during inference), and FPS (the number of image frames processed per second).

D. Ablation Study

Ablation experiments were performed on the BSD dataset from both qualitative and quantitative perspectives to explore the role of each component in the framework's performance.

1) The Influence of FGEM. FGEM improves the extraction of fine-grained features by utilizing the feature maps from adjacent stages in the Backbone, particularly for small and low-separability defects. To validate its effectiveness, we removed FGEM from FHS-Net and utilized a 1×1 convolution for dimensional alignment, defining this model as w/o_FGEM. To further dissect the internal rationale of FGEM, we constructed two fine-grained variants, namely w/o_Adjacent and

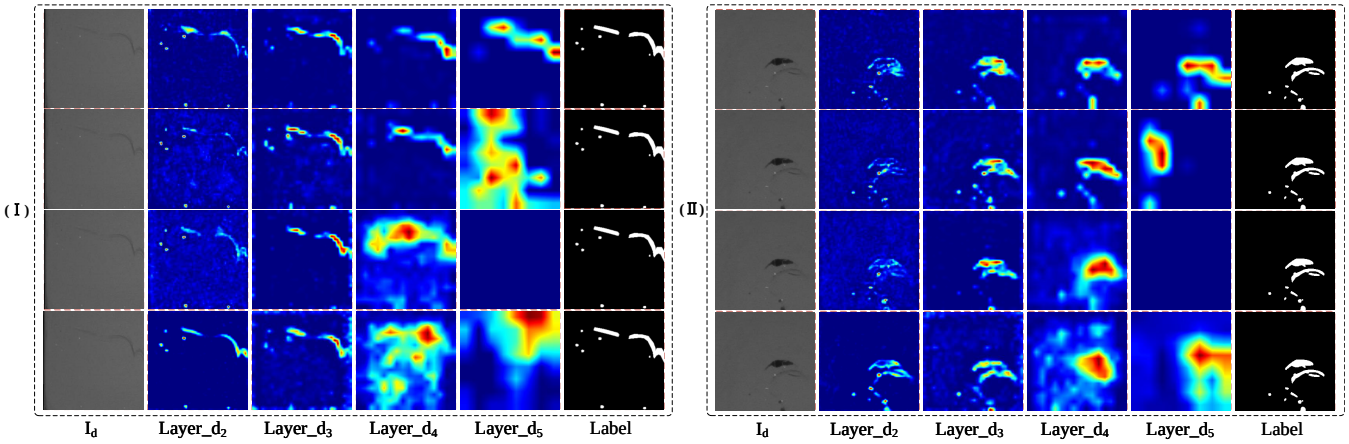

Fig. 9. Quantitative analysis results of different layers: Layer_d2, Layer_d3, Layer_d4, and Layer_d5 based on Grad-CAM. Two sets of images, (I) and (II), were selected from the BSD test set. A total of four rows of images are included, with each row representing FHS-Net, FHS-Net without Mask5 supervision, FHS-Net without Mask4&5 supervision, and FHS-Net without Mask3&4&5 supervision, respectively. The redder the image, the more attention the model pays to it. HPD is employed to address the issue of vanishing gradients and to enhance the middle layer's ability to detect defective regions in the network.

w/o_Attention, where the former removes inputs from adjacent stages to verify multi-scale integration, and the latter eliminates the channel weighting branch to test feature highlighting. We similarly constructed the w/o_FGEM_HPD variant based on the architecture without HPD. Furthermore, to demonstrate the structural superiority of FGEM over other designs, we extended the experiment by substituting FGEM with three representative feature fusion and attention modules, including Adaptively Spatial Feature Fusion (ASFF) [36], Adaptive Multiscale Feature Enhancement (AME) [45], and the Convolutional Block Attention Module (CBAM) [46].

From a quantitative perspective, as shown in Part (a) of Table II, FHS-Net achieves an F1 score improvement of 2.31 and an IoU improvement of 2.57 over w/o_FGEM, confirming the effectiveness of FGEM in refining detail-level features. To further assess the influence of module combinations, we removed FGEM from the variant w/o_HPD, resulting in w/o_FGEM_HPD. This model underperforms w/o_HPD, with an F1 score drop of 1.04 and IoU drop of 1.13. The consistent degradation observed in different architectural variants reinforces the critical importance of FGEM. Furthermore, regarding the internal design analysis in Part (b), experimental results demonstrate that w/o_Adjacent and w/o_Attention exhibit F1 score drops of 1.06 and 1.09 compared to FHS-Net, respectively. This confirms that both the multi-scale integration and channel attention mechanism are essential for refining defect features. Regarding the replacement analysis in Part (b), experimental results indicate that while these alternative modules improve performance compared to the baseline, FHS-Net equipped with FGEM consistently yields the highest accuracy. This suggests that FGEM's specific design for preserving adjacent-stage consistency is more effective for the Siamese difference extraction task than general-purpose modules, establishing it as the optimal choice.

From a qualitative analysis perspective, we selected four groups of images from the BSD test set and visualized the differences in feature layers using GradCAM [47]. We designated these layers as Layer_d2 - Layer_d5. The visualization results are presented in Fig. 8, where each group of images consists of two rows: the first row displays the results from FHS-Net, while the second row shows the results from w/o_FGEM. The removal of FGEM reduces the model's capacity to capture fine-grained features. For instance, in image (I), a low-separability scratch and small dust particles are visible on the surface of the battery. In Layer_d2, FHS-Net is able to focus more on the edge information and fine-grained details of the defect compared to w/o_FGEM. In Layer_d3, w/o_FGEM fails to concentrate on the defect information and instead emphasizes irrelevant background details, demonstrating that FGEM not only enriches spatial detail but also suppresses misleading activations.

TABLE II
ABLATION STUDY ON DIFFERENT COMPONENTS

Model Variant	F1(%)	IoU(%)	Params(M)	FLOPs(G)
(a) Component Ablation				
w/o_Sia	64.54	47.64	3.17	0.86
w/o_FGEM	64.85	47.98	2.50	1.13
w/o_HPD	64.97	48.11	3.03	2.09
w/o_FGEM_HPD	63.93	46.98	**2.37**	0.88
w/o_FGEM_HPD_Sia	63.50	46.52	**2.37**	**0.44**
(b) FGEM				
w/o_ Adjacent	66.10	49.54	2.89	1.72
w/o Attention	66.07	49.51	3.17	2.35
ASFF (Replacement)	65.95	49.19	2.76	1.92
AME (Replacement)	65.38	48.57	3.18	2.23
CBAM (Replacement)	65.97	49.22	3.10	1.79
(c) HPD				
FPN (Replacement)	65.04	48.19	3.05	2.31
PANet (Replacement)	65.29	48.47	3.27	2.41
U-Net++ (Replacement)	66.13	49.40	3.73	4.20
w/o_alignment	66.06	49.32	3.17	2.35
FHS-Net (Ours)	**67.16**	**50.55**	3.17	2.35

2) The influence of HPD. HPD incorporates loss computation at each decoding layer to establish a hierarchical perception mechanism. This approach enables the model to begin perceiving defects at intermediate layers, thereby progressively enhancing detection accuracy across the network.

Each stage undergoes fine-tuning, which improves the detection capabilities of the intermediate layers for defect regions. To validate its effectiveness, we developed a decoder variant named w/o_HPD, in which the HPB is replaced with a 3×3 convolution. This variant maintains the hierarchical structure of HPD by generating four intermediate mask maps in the decoder, which are then bilinearly upsampled to the label size, summed, and passed through a sigmoid function for loss computation. Based on w/o_HPD, we further removed the FGEM component using the same strategy as in w/o_FGEM, resulting in a new variant referred to as w/o_FGEM_HPD. Furthermore, to demonstrate the structural superiority of HPD over other designs, we extended the experiment by substituting HPD with three representative multi-scale feature aggregation architectures, including Feature Pyramid Network (FPN) [48], Path Aggregation Network (PANet) [49], and U-Net++ [50]. We then systematically compared the performance of these variants to assess both the necessity and replaceability of our proposed decoder.

Based on quantitative analysis, Part (a) of Table II indicates that FHS-Net outperforms w/o_HPD, achieving a 2.19 increase in F1-score and a 2.44 increase in IoU. Additionally, w/o_FGEM demonstrates better performance than w/o_FGEM_HPD, with a 0.92 increase in F1-score and a 1.0 increase in IoU. These comparisons suggest that HPD not only enhances feature decoding independently but also works synergistically with FGEM to further improve segmentation accuracy. Regarding the replacement analysis in Part (c), experimental results demonstrate that among all compared decoder architectures, FHS-Net equipped with HPD achieves the optimal performance, validating its superiority over generic multi-scale architectures. Furthermore, we conducted an ablation study by selectively applying the loss supervision to different hierarchical mask outputs. As shown in Table III, each row represents a different variant where the loss is computed on specific masks (✓) while others are ignored (×). As shown in the first four rows, the performance follows the order Mask2 > Mask3 > Mask4 > Mask5. This is because each HPB module refines features and passes information to the previous stage. When only Mask2 is supervised, it indirectly benefits from the refinements of all preceding modules, leading to better results. In other words, HPB2 makes a greater contribution to the final performance. Furthermore, we supervise the outputs of all four HPB stages simultaneously, which achieves the best performance. This is because the hierarchical supervision enables intermediate layers to perceive defects earlier during decoding, allowing the model to refine features more effectively at each stage.

In terms of qualitative analysis, two sets of images were selected from the BSD test set, and GradCam was employed to visualize the subtracted feature layers, specifically Layer_d2 to Layer_d5. The visualization results are presented in Fig.9, where each set of images consists of four rows representing FHS-Net, without Mask5 supervision, without Mask4&5 supervision, and without Mask3&4&5 supervision, respectively. The objective is to investigate the impact of different perceptual strategies on the model's internal representation. When the loss is computed solely for Mask2, Mask3, and Mask4 (i.e., without Mask5 supervision), the model fails to detect defect regions in the intermediate Layer_d5 (second row, fifth column in both sets), as Mask5 has a shorter backpropagation path to Layer_d5. In contrast, when supervision is applied to Mask5, it facilitates more effective training for Layer_d5, thereby enhancing its ability to localize defect regions. Both the model without Mask4&5 supervision and the model without Mask3&4&5 supervision show reduced attention to defect regions in Layer_d4 and Layer_d5. Notably, the model trained without Mask3&4&5 supervision also introduces increased background noise in Layer_d3 (fourth row, third column in both sets). These results further demonstrate that HPD alleviates the gradient vanishing problem and enhances the network's capacity to detect defects in intermediate feature layers.

3) The Influence of Siamese Structure. Our proposed framework accepts both a target image and a reference image, leveraging the differences between the two to identify defects. To assess the effectiveness of this structure, we removed the reference branch from FHS-Net, retaining only the target image input, and designated this modified model as w/o_Sia. Similarly,

TABLE III
ABLATION STUDY ON HPD DECODER

mask2	mask3	mask4	mask5	F1(%)	IoU(%)	Parameter(M)	FLOPs(G)
×	×	×	✓	50.72	33.98	3.17	2.34
×	×	✓	×	58.24	41.08	3.17	2.34
×	✓	×	×	62.90	45.88	3.17	2.34
✓	×	×	×	65.08	48.24	3.17	2.34
✓	✓	×	×	66.20	49.47	3.17	2.34
✓	✓	✓	×	65.89	49.13	3.17	2.34
✓	✓	✓	✓	**67.16**	**50.55**	3.17	2.34

TABLE IV
COMPARISON OF DIFFERENT FEATURE FUSION STRATEGIES

Fusion Strategy	F1(%)	IoU(%)	Params(M)	FLOPs(G)
Add	65.80	49.03	3.17	2.35
Concat	66.20	49.47	3.20	2.39
Cos-Sim	65.06	48.21	**3.17**	**2.35**
Abs-Diff	**67.16**	**50.55**	**3.17**	**2.35**

TABLE V
SENSITIVITY ANALYSIS OF LOSS WEIGHT λ

λ	F1(%)	IoU(%)	λ	F1(%)	IoU(%)
0.0	63.42	46.44	**1.0**	**67.16**	**50.55**
0.2	64.68	47.79	1.2	67.07	50.46
0.4	64.21	47.29	1.4	64.86	48.00
0.6	66.53	49.84	1.6	66.62	49.94
0.8	65.16	48.33	1.8	64.97	48.11

TABLE VI
PERFORMANCE COMPARISION ON DIFFERENT BACKBONE

Backbone	F1(%)	IoU(%)	Params(M)	FLOPs(G)	FPS
Resnet18	63.87	46.91	16.96	26.31	67.77
Resnet34	65.64	48.86	27.07	47.48	49.74
Resnet50	66.39	49.69	43.80	54.26	34.25
MobileNetV1	65.14	48.30	4.64	3.08	**81.64**
MobileNetV2	**67.16**	**50.55**	**3.17**	2.347	80.95
MobileNetV3	64.32	47.41	3.49	**1.66**	73.35

we eliminated the reference branch from w/o_FGEM_HPD, resulting in a model referred to as w/o_FGEM_HPD_Sia. The experimental results presented in Table II demonstrate that FHS-Net outperforms w/o_Sia, with a 2.62 increase in F1 and a 2.91 increase in IoU. Likewise, w/o_FGEM_HPD outperforms w/o_FGEM_HPD_Sia, achieving a 0.43 increase in F1 and a 0.46 increase in IoU. These results suggest that incorporating a reference image significantly enhances network's capacity to distinguish between normal and defective regions on the battery surface, particularly when identifying low-separability or subtle defects.

4) The Influence of Feature Fusion Strategies. FHS-Net leverages a reference-based design to localize defects by analyzing the discrepancy between the inspection and reference images. We evaluated the proposed Absolute Difference (Abs-Diff) against Element-wise Addition (Add), Concatenation (Concat), and Cosine Similarity (Cos-Sim). The quantitative results in Table IV. Specifically, compared with Abs-Diff, Concat results in a decrease of 0.96 in F1-score and 1.08 in IoU, as it forces the subsequent layers to implicitly learn the subtraction logic, increasing optimization difficulty. Furthermore, Cos-Sim results in a decrease of 2.10 in F1-score and 2.34 in IoU, as it prioritizes directional alignment over the difference magnitude between the inspection and reference images. In contrast, the Abs-Diff operator aligns perfectly with our design premise by explicitly computing the intensity of discrepancies, acting as a robust prior for defect localization.

5) The Influence of loss weight λ. To determine the optimal weight λ in Equation 20 that balances L_{ce} and L_{dice}, we conducted a sensitivity analysis by varying λ from 0.0 to 1.8, as shown in Table V. The results demonstrate that the model achieves its peak performance at $\lambda = 1$, reaching the highest F1 and IoU. Specifically, relying solely on L_{ce} where λ equals 0 results in a significant decrease of 3.74 in F1 and 4.11 in IoU, highlighting the necessity of L_{dice} in addressing the class imbalance between defects and background. Conversely, excessively increasing the weight such as setting λ to 1.8 leads to diminishing returns, indicating that over-emphasizing region-based supervision may disrupt the optimization balance. Consequently, we set $\lambda = 1$ as the default hyperparameter for all reported experiments.

6) The Influence of different backbone: Various backbones, such as ResNet18, ResNet34, ResNet50, MobileNetV1, MobileNetV2 and MobileNetV3 were tested in FHS-Net to identify the optimal backbone for battery surface defect detection and assess their impact on our framework. As indicated in Table VI, although the deeper ResNet50 outperforms ResNet34, it fails to yield a significant performance leap over the lightweight MobileNetV2. This suggests that the massive parameter capacity of heavyweight models is less effective for these specific texture features when trained on the BSD dataset. While the BSD dataset scale of 2,183 sub-images is comparable to established benchmarks like NEU-SEG [40] and Magnetic-Tile-Defect [41], the inherent scarcity of diverse anomaly samples in industrial production environments limits the generalization advantage of excessively deep architectures. In contrast, MobileNetV2 strikes a more favorable balance between feature abstraction and spatial detail preservation, as its compact structure is inherently better suited for capturing low-contrast defect patterns without the signal dilution often found in deeper networks. Table VI proved that MobileNetV2 strikes a more favorable balance between accuracy and efficiency, making it the most appropriate choice for FHS-Net, especially in real-time industrial scenarios where both performance and speed are critical.

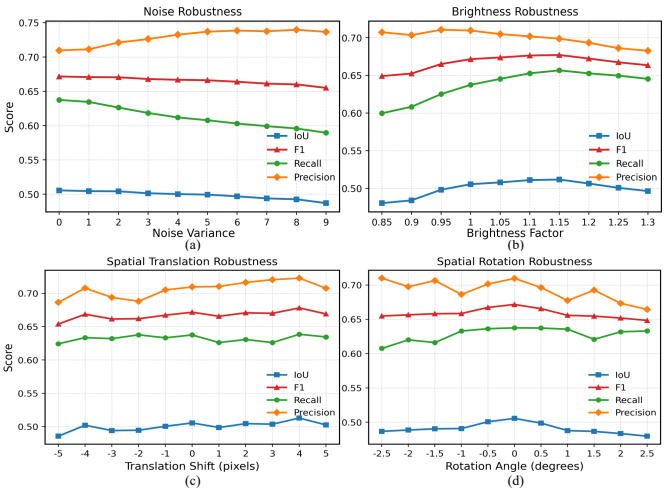

Fig. 10. Robustness analysis of the reference image against (a) Gaussian noise, (b) brightness variations, (c) spatial translation shifts, and (d) spatial rotation angles.

7) Robustness analysis. To evaluate the model's reliability, we conducted a sensitivity analysis by introducing various perturbations during inference, as illustrated in Fig. 10. In Figs. 10(a) and (b), we first examine the image quality perturbations applied to the reference branch. In Fig. 10(a), performance metrics exhibit a high degree of stability despite increasing noise levels, demonstrating that the model's feature extraction capability is resilient to image noise. Similarly, Fig. 10(b) shows consistent detection accuracy across brightness scaling factors ranging from 0.85 to 1.3 times the original intensity. Although a marginal decrease occurs at extremes, the overall trend confirms the model's resilience to illumination changes. Furthermore, to assess the model's tolerance to spatial misalignments during the inspection process, we introduced spatial perturbations to the query images in Figs. 10(c) and (d). Fig. 10(c) illustrates the impact of spatial translation shifts within a range of [-5, 5] pixels. The remarkably stable curves indicate that the proposed network can effectively handle minor positional offsets between the reference and query images. Finally, Fig. 10(d) evaluates the sensitivity to spatial rotation angles within a range of $[-2.5°, 2.5°]$. The negligible fluctuations in all metrics demonstrate the model's spatial invariance and reliability, ensuring robust defect detection even when the industrial components are not perfectly aligned with the template.

8) The Influence of the alignment algorithm. To verify the necessity of the proposed alignment for mitigating spatial discrepancies between the query image and the defect-free template, we conducted an ablation experiment by disabling this component, as shown in Table II. Without the align-

TABLE VII
PERFORMANCE COMPARISON ON F1 (%), IOU (%), PRECISION (%), RECALL (%), PARAMS (M), GFLOPS, GPU MEMORY, AND FPS.

Method	backbone	BSD				NEU				MTD				Speed			
		F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	Params	GFLOPs	GPU Mem	FPS
BiFormer	Biformer-B	65.81	49.04	74.62	58.86	91.89	84.99	92.30	91.48	**92.74**	**86.46**	92.18	93.30	87.54	65.72M	2160 Mib	32.62
PoolFormer	PoolFormer-s12	62.21	45.15	**77.84**	51.80	89.96	81.75	90.39	89.53	89.25	80.58	91.31	87.28	15.67	7.76G	482 Mib	134.31
ConvNeXt	ConvNeXt-T	65.73	48.95	75.99	57.91	90.76	83.08	91.53	90.00	90.14	82.06	87.83	92.59	120.89	73.20G	856 Mib	38.15
SegNeXt	MSCAN-T	64.65	47.77	76.53	55.97	91.14	83.72	91.63	90.66	90.27	82.27	86.51	94.38	4.26	**1.61G**	408 Mib	166.72
Mask2Former	ResNet50	61.63	44.54	70.97	54.57	89.73	81.37	90.01	89.45	89.50	81.00	88.01	91.05	44.04	-	666 Mib	36.97
Mask2Former	MobileNetV2	63.93	46.99	70.28	58.64	89.15	80.43	88.80	89.51	89.65	81.25	90.75	88.58	21.44	-	500 Mib	47.01
SegFormer	MiT-B0	64.00	47.06	68.68	59.92	91.11	83.66	91.73	90.48	87.36	77.56	79.87	**96.42**	3.75	1.93G	424 Mib	96.25
MaskFormer	ResNet50	65.46	48.65	74.57	58.33	90.00	81.82	90.99	89.03	90.56	82.74	89.89	91.23	41.31	-	638 Mib	54.83
BiSeNetV2	Bisenetv2	51.10	34.32	55.01	47.71	88.00	78.57	87.36	88.65	84.74	73.52	90.16	79.93	3.36	3.09G	**352 Mib**	**317.69**
CCNet	ResNet50	62.21	45.15	73.71	53.82	89.37	80.78	90.10	88.65	89.52	81.03	88.81	90.24	47.53	50.29G	664 Mib	62.83
CCNet	MobileNetV2	55.26	38.18	74.14	44.04	88.00	78.57	89.41	86.64	88.77	79.81	91.15	86.51	9.82	10.57G	466 Mib	184.73
DeepLabv3+	ResNet50	63.98	47.04	73.38	56.71	90.02	81.86	90.64	89.42	91.77	84.80	92.56	91.00	44.22	44.12G	620 Mib	66.25
DeepLabv3+	MobileNetV2	55.89	38.79	77.77	43.62	88.95	80.10	90.30	87.64	88.95	80.10	90.30	87.64	15.01	17.05G	404 Mib	138.96
PSPNet	ResNet50	61.97	44.90	76.21	52.22	89.46	80.93	90.25	88.69	91.18	83.92	91.18	91.33	46.68	44.78G	624 Mib	62.63
PSPNet	MobileNetV2	57.83	40.68	69.92	49.31	89.90	81.65	90.95	88.87	90.79	83.13	90.33	91.25	13.94	13.16G	358 Mib	176.79
FCN	ResNet50	61.12	44.01	76.65	50.83	89.17	80.45	90.13	88.23	91.15	83.51	91.15	90.88	47.20	49.54G	654 Mib	56.78
BG-Net	ResNet34	64.93	48.07	74.09	57.79	91.92	85.05	91.80	92.05	89.67	81.28	89.88	89.47	47.69	16.48G	5322 Mib	34.30
FC-diff	-	65.42	48.61	71.90	60.01	90.08	81.95	90.01	90.16	87.22	77.33	89.17	85.35	**1.35**	4.72G	1122 Mib	83.81
TFI-Net	ResNet18	65.70	48.92	69.42	62.35	91.80	84.84	90.93	92.68	90.33	82.37	90.84	89.83	28.37	9.74G	950 Mib	64.44
FHS-Net(Ours)	MobileNetV2	**67.16**	**50.55**	70.95	**63.75**	**92.73**	**86.44**	**92.70**	**92.76**	91.55	84.41	**95.86**	87.61	3.17	2.35G	622 Mib	80.95

ment algorithm, the model's performance on the test set exhibits a decline, with the IoU falling to 49.32 and the F1 score dropping to 66.06. This performance degradation is primarily attributed to pixel-level spatial offsets caused by mechanical positioning errors in actual production lines. Such misalignments cause the background textures to be incorrectly identified as anomalies during the feature differencing stage.

E. Comparative Experiments

We Conduct a comparison of the proposed FHS-Net with other advanced models using three defect detection datasets. These models include BiFormer [51], PoolFormer [52], ConvNeXt [53], SegNeXt [54], Mask2Former [55], SegFormer [56], MaskFormer [57], BiSeNetV2 [58], CCNet [59], DeepLabv3+ [60], PSPNet [61] and FCN [62]. In addition, we also introduce three recent models that employ differencing strategies into our benchmark. BG-Net [31] is a defect detection method specifically designed for industrial inspection, which utilizes background-guided differencing to enhance abnormal region localization. Although FC-diff [42] and TFI-Net [43] originate from remote sensing change detection, their structural designs—such as dual-branch representations and difference-aware fusion—are conceptually related to industrial scenarios, where distinguishing subtle differences between normal and defective samples is essential. Furthermore, we replaced the original backbones of several SOTA methods (Mask2Former [55], CCNet [59], DeepLabv3+ [60], and PSPNet [61]) with MobileNetV2, which is also used in our FHS-Net. This uniform backbone setting enables a fairer comparison of the architectural differences among these methods. The results of the experiments are presented in Tables VII. The optimal results are indicated in bold, the second-best results are underlined, and the third-ranked results are marked with a wavy line.

1) Comparison of the BSD dataset. Table VII shows the comparison results of FHS-Net against 19 baseline models. The proposed FHS-Net achieves the highest detection accuracy, with F1: 67.16, IoU: 50.55. Compared to BiFormer, the F1 improves by 1.35, and the IoU improves by 1.51. We selected two sets of images from the test set of the BSD dataset for visualization. While all existing baselines can approximately locate the defects, some issues remain. Qualitatively, the results are shown in Fig. 11. In the Group I images, there is a depression that has a contrast similar to the background color. Most baseline models struggle with missed detection when encountering defects with low separability (e.g., PoolFormer). In the Group II images, there is a scratch that also presents a similar contrast, posing the same challenge of low visibility, leading to missed detections in most baseline models (e.g., SegFormer). In contrast, our proposed model effectively localizes the edges of the defects, even in the presence of low separability and small defect sizes. In the Group III images, there are two fine dust particles on the surface of the battery, accompanied by small defects. Most baseline models struggle with misdetection and are unable to accurately localize the edges of these defects (e.g., Biformer). However, benefiting from the Fine-Grained Enhancement Module, our proposed framework can precisely identify the defects by utilizing edge information.

2) Comparison of the NEU-SEG dataset [40]. Table VII shows the comparison results of FHS-Net against 19 baseline models. The proposed FHS-Net achieves optimal detection accuracy, F1: 92.73, IoU: 86.44. Compared to BG-Net, the F1 improves by 0.81, and the IoU improves by 1.39. We selected two sets of images from the test set of the NEU-SEG dataset for visualization. Qualitatively, the results are shown in Fig. 11. Existing baselines exhibit varying degrees of leakage and misdetection issues when confronted with different scenarios. In Group I images, there are low-separability patches on the

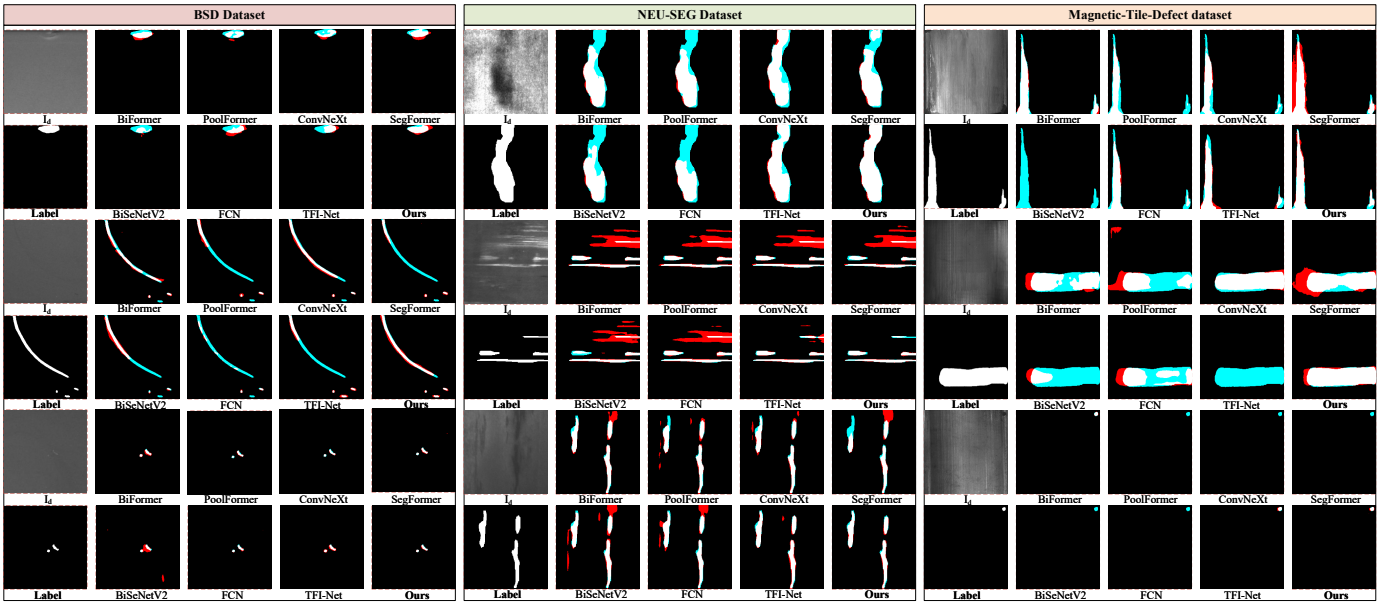

Fig. 11. Qualitative comparison results of FHS-Net and other baselines on the test sets of BSD, NEU-SEG, and Magnetic-Tile-Defect datasets. White indicates true positives, red indicates false positives, black indicates true negatives, and blue indicates false negatives.

steel surface, which present a low-separability challenge. Most of the baseline models have issues with missed detections, however, our model is able to accurately locate the defects. In Group II images, there are tiny scratches that are surrounded by a similar background. Many models experience misdetection issues, however, our model effectively avoids this drawback. In the Group III images, there are low-separability inclusions on the steel surface. When faced with the low contrast between the background and the defective region, some models suffer from misdetection (e.g., BiFormer), while others struggle with missed detection (e.g., PoolFormer).

3) Comparison on the Magnetic-Tile-Defect dataset [41]. Table VII shows the comparison results of FHS-Net against 19 baseline models. The proposed FHS-Net achieves the third-best detection accuracy, with an F1 score of 91.55 and an IoU of 84.41. When compared to BiFormer, the difference in the F1 score is only 1.19, and the difference in IoU is 2.05. However, FHS-Net has the lowest Params and lower FLOPs, while BiFormer has 27.62 times more parameters and 27.97 times higher FLOPs compared to FHS-Net. For visualization, we selected two sets of images from the test set of the Magnetic-Tile-Defect dataset. Qualitatively, the results are shown in Fig. 11. The existing baselines exhibit varying degrees of under-detection issues when confronted with different scenarios. In the Group I images, there is a break with low separability on both sides of the tile, causing most models to suffer from missed detections (e.g., BiSeNetV2), and some to experience misdetection (e.g., SegFormer). In the Group II image, there is a low-separability uneven area beneath the surface of the magnetic tile. Most models exhibit missed detections (e.g., BiFormer), and some fail to accurately locate the edges of the defects, leading to misdetections (e.g., SegFormer). However, our proposed framework effectively addresses these challenges. In the Group III images, there is a tiny Blowhole in the upper right corner of the tile. Most models experience missed detections (e.g., Segformer), while our proposed framework successfully avoids this issue.

4) Discussion. To analyze the underlying reasons for FHS-Net's strong performance across the BSD datasets, NEU-SEG [40] and Magnetic-Tile-Defect [41], we summarize its key architectural advantages as follows. First, FHS-Net employs a Siamese architecture to explicitly model inter-image discrepancies, outperforming traditional single-stream architectures in complex scenarios [51]–[62]. Second, the FGEM leverages multi-stage neighboring features to enhance the representation of small or low-contrast defects. This design addresses the limitations of methods that primarily focus on high-level features, which often overlook fine-grained details and shallow spatial information [42], [51], [56], [57], [59]. Third, the HPD introduces progressive semantic reconstruction using multi-scale masks, capturing both coarse and fine features. This approach alleviates semantic dilution commonly encountered in shallow decoders or single-path architectures, a limitation observed in prior works [42], [56], [59]–[62]. Finally, FHS-Net's fully convolutional design ensures efficient inference without compromising accuracy. While Transformer-based models often sacrifice efficiency for accuracy [31], [51], [55]–[57], FHS-Net achieves a more favorable balance, making it well-suited for real-time industrial inspection tasks. Notably, this balance is especially advantageous for datasets of small to medium scale, where convolutional architectures with stronger inductive biases often yield better generalization. Moreover, our BSD dataset yields relatively lower performance. Many defects arise from internal damage and exhibit only subtle surface cues, often blending into complex textures with low contrast and high variability, making detection particularly difficult, which highlights its inherent challenges.

Furthermore, a comparison of model scales reveals that the performance gains are driven by architectural innovation rather than the scale of model parameters. As shown in Table

VII, FHS-Net achieves superior performance with only 3.17 M params and 2.35 GFLOPs. It significantly outperforms heavyweight models such as BiFormer [51] (87.54 M params, 65.72 GFLOPs) and ConvNeXt [53] (120.89 M params, 73.20 GFLOPs) despite their much larger capacities. Even compared to lightweight designs, our framework achieves a more favorable balance between accuracy and complexity. For instance, while FC-diff [42] possesses a smaller footprint of 1.35 M parameters, FHS-Net provides higher detection accuracy. Similarly, although SegNeXt [54] (4.26 M params, 1.61 GFLOPs) and SegFormer (3.75 M params, 1.93 GFLOPs) offer competitive complexity, their performance remains inferior across all evaluated surface defect detection benchmarks.

VI. Limitations and Future Work

In this work, we focus on detecting battery surface defects. Our reference-based discrepancy-aware architecture is particularly suitable for highly standardized production lines. In the future, we plan to extend the proposed discrepancy-aware architecture to other key stages of battery manufacturing, including welding spot defect detection. Meanwhile, although our current approach adopts a binary segmentation strategy, many defects in the BSD dataset originate from internal structural issues and show up as subtle surface signs—some of which are difficult to detect using RGB images alone. To address this limitation, we plan to enhance the dataset by incorporating depth images, which can complement RGB inputs by providing topological information to highlight otherwise indistinct defect patterns. Based on this enhancement, future work will explore multi-class semantic segmentation for more fine-grained defect categorization. Furthermore, our research currently focuses on enhancing defect segmentation accuracy and robustness using a fully supervised framework that depends on large volumes of labeled data. To alleviate such data dependency, future work will investigate self-supervised and unsupervised learning approaches tailored to battery surface defect detection.

VII. Conclusion

In this study, we collected and annotated the Battery Surface Defect dataset to advance the field of surface defect segmentation. We also propose a Fine-Grained Hierarchical Perception Siamese Network to address challenges posed by the complex and variable shapes of defect samples and the low contrast between defects and backgrounds. Additionally, we performed comprehensive experiments using the BSD, NEU-SEG, and Magnetic-Tile-Defect datasets. Our model achieved optimal accuracy on the BSD and NEU-Seg datasets and ranked third on the Magnetic-Tile-Defect dataset, all while maintaining low model parameter count, computational efficiency, and real-time performance.

References

[1] O. Badmos, A. Kopp, T. Bernthaler, and G. Schneider, "Image-based defect detection in lithium-ion battery electrode using convolutional neural networks," *J. Intell. Manuf.*, vol. 31, no. 4, pp. 885–897, Apr. 2020.

[2] J. Xu, Y. Liu, H. Xie, and F. Luo, "Surface quality assurance method for lithium-ion battery electrode using concentration compensation and partiality decision rules," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 6, pp. 3157–3169, Jun. 2020.

[3] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016.

[4] Y. Li, W. Zhao, and J. Pan, "Deformable patterned fabric defect detection with fisher criterion-based deep learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1256–1264, Apr. 2017.

[5] D. Chetverikov and A. Hanbury, "Finding defects in texture using regularity and local orientation," *Pattern Recognit.*, vol. 35, no. 10, pp. 2165–2180, Oct. 2002.

[6] S. B. Jha and R. F. Babiceanu, "Deep cnn-based visual defect detection: Survey of current literature," *Comput. Ind.*, vol. 148, p. 103911, 2023.

[7] S. Zhuang, H. Zhang, D. Liang, H. Liu, and Z. Gao, "Adaptive sequential bayesian iterative learning for myocardial motion estimation on cardiac image sequences," *IEEE Trans. Med. Imaging*, 2025, early Access.

[8] R. Ren, T. Hung, and K. C. Tan, "Automatic microstructure defect detection of ti-6al-4v titanium alloy by regions-based graph," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 2, pp. 87–96, 2017.

[9] T. Meng, Y. Tao, Z. Chen, J. R. S. Avila, Q. Ran, Y. Shao, R. Huang, Y. Xie, Q. Zhao, Z. Zhang *et al.*, "Depth evaluation for metal surface defects by eddy current testing using deep residual convolutional neural networks," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[10] G. Fu, P. Sun, W. Zhu, J. Yang, Y. Cao, M. Y. Yang, and Y. Cao, "A deep-learning-based approach for fast and robust steel surface defects classification," *Opt. Lasers Eng.*, vol. 121, pp. 397–405, 2019.

[11] T. Wang, Z. Li, Y. Xu, J. Chen, A. Genovese, V. Piuri, and F. Scotti, "Few-shot steel surface defect recognition via self-supervised teacher-student model with min-max instances similarity," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–16, 2023.

[12] Y. Xu, J. Chen, Y. Liang, Y. Zhai, Z. Ying, W. Zhou, A. Genovese, V. Piuri, and F. Scotti, "Flexible and diverse contrastive learning for steel surface defect recognition with few labeled samples," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–14, 2023.

[13] Z. Cao, K. Chen, J. Chen, Z. Chen, and M. Zhang, "Cacs-yolo: A lightweight model for insulator defect detection based on improved yolov8m," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–10, 2024.

[14] J. Bao, X. Yuan, Q. Wu, C.-T. Lam, W. Ke, and P. Li, "Cca-yolo: Channel and coordinate aware-based yolo for photovoltaic cell defect detection in electroluminescence images," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–12, 2025.

[15] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019.

[16] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020.

[17] Y. Zhong, Z. Ling, L. Liu, S. Zhang, and H. Wen, "Deep gaussian attention network for lumber surface defect segmentation," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–12, 2024.

[18] F. Yan, X. Jiang, Y. Zhang, Y. Lu, X. Nan, S. He, and M. Xu, "Progressive feature enhancement network for surface defect segmentation," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. early access, pp. 1–11, 2025.

[19] C. Zhang, Y. Qi, and Y. Wang, "Learning the color space for effective fabric defect detection," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 8, no. 1, pp. 981–991, 2024.

[20] Y. Zhou, X. Xu, J. Song, F. Shen, and H. T. Shen, "Msflow: Multiscale flow-based framework for unsupervised anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2024.

[21] V. Zavrtanik, M. Kristan, and D. Skočaj, "Reconstruction by inpainting for visual anomaly detection," *Pattern Recognit.*, vol. 112, p. 107706, 2021.

[22] J. Zhu, P. Yan, J. Jiang, Y. Cui, and X. Xu, "Asymmetric teacher–student feature pyramid matching for industrial anomaly detection," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, 2024.

[23] S.-J. Peng, Y. Fan, Y.-m. Cheung, X. Liu, Z. Cui, and T. Li, "Towards efficient cross-modal anomaly detection using triple-adaptive network and bi-quintuple contrastive learning," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 8, no. 1, pp. 697–709, 2024.

[24] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *Adv. Neural Inf. Process. Syst.*, vol. 6. Morgan Kaufmann, 1993, pp. 737–744.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[28] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.

[29] C. Wang, A. Bochkovskiy, and H. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2023, pp. 7464–7475.

[30] H. Zhu, J. You, Y. Zhai, Y. Xu, T. Wang, Y. Chen, J. Zhou, P. Coscia, A. Genovese, and C. L. P. Chen, "Ds2a-former: Battery surface defect segmentation via dual-stream spatial attention transformer network," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–17, 2025.

[31] B. Chen, T. Niu, R. Zhang, H. Zhang, Y. Lin, and B. Li, "Feature matching driven background generalization neural networks for surface defect segmentation," *Knowl.-Based Syst.*, vol. 287, p. 111451, 2024.

[32] V. Reshadat and R. A. J. W. Kapteijns, "Improving the performance of automated optical inspection (aoi) using machine learning classifiers," in *2021 Int. Conf. Data Softw. Eng. (ICoDSE)*, Nov. 2021, pp. 1–5.

[33] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[34] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 202–211.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[36] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," *arXiv preprint arXiv:1911.09516*, Nov. 2019.

[37] Z. Fu, J. Li, Z. Hua, and L. Fan, "Deep supervision feature refinement attention network for medical image segmentation," *Eng. Appl. Artif. Intell.*, vol. 125, p. 106666, Oct. 2023.

[38] P. T. D. Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.*, vol. 134, no. 1, pp. 19–67, Feb. 2005.

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

[40] Y. Bao, K. Song, J. Liu, Y. Wang, Y. Yan, H. Yu, and X. Li, "Triplet-graph reasoning network for few-shot metal generic surface defect segmentation," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[41] Y. Huang, C. Qiu, Y. Guo, X. Wang, and K. Yuan, "Surface defect saliency of magnetic tile," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 612–617.

[42] R. C. Daudt, B. L. Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 2018, pp. 4063–4067.

[43] Z. Li, C. Tang, L. Wang, and A. Y. Zomaya, "Remote sensing change detection via temporal feature interaction and guided refinement," *IEEE Trans. Geosci. Remote. Sens.*, vol. 60, pp. 1–11, 2022.

[44] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *Int. J. Comput. Vis.*, vol. 127, pp. 302–321, 2019.

[45] Z. Ying, Y. Zhou, Y. Zhai, H. Zhu, H. Zhang, P. Coscia, A. Genovese, F. Scotti, V. Piuri, and C. L. P. Chen, "Aegl-net: Adaptive multiscale global-local feature fusion network for remote sensing change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–14, 2025.

[46] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[48] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[49] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8759–8768.

[50] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learn. Med. Image Anal. (DLMIA)*. Springer, Sep. 2018, pp. 3–11.

[51] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. Lau, "BiFormer: Vision Transformer with Bi-Level Routing Attention," in *2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, Jun. 2023.

[52] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2022, pp. 10 809–10 819.

[53] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2022, pp. 11 966–11 976.

[54] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "Segnext: Rethinking convolutional attention design for semantic segmentation," in *Adv. Neural Inf. Process. Syst. (NeurIPS) 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022.

[55] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) 2022*. IEEE, 2022, pp. 1280–1289.

[56] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Álvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2021, pp. 12 077–12 090.

[57] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Adv. Neural Inf. Process. Syst. (NeurIPS) 2021*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 17 864–17 875.

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

[59] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 6568–6577.

[60] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. Springer, 2018, pp. 833–851.

[61] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2017, pp. 2881–2890.

[62] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.