DS^{2}A-Former: Battery Surface Defect Segmentation via Dual Stream Spatial Attention Transformer Network

Hufei Zhu*, Jie You*, Yikui Zhai*, *Senior Member, IEEE*, Ying Xu, Tianlei Wang, Yingwen Chen, Jianhong Zhou, Pasquale Coscia, *Senior Member, IEEE*, Angelo Genovese, *Senior Member, IEEE*, C. L. Philip Chen, *Fellow, IEEE*

***Abstract*—With the advancement of industrial automation, surface defect inspection has become crucial for battery manufacturing quality assurance. However, existing public datasets lack sufficient resources for battery surface defects, hindering the process of battery defect inspection. To address this, we developed the Multi-class Battery Surface Defect dataset (MBSD), which contains 2,491 images and 6,639 samples spanning three defect types. Meanwhile, deep learning-based methods have made progress in industrial quality inspection but face challenges in handling complex battery defect details, defect category similarities, and small target defects. These issues hinder accurate, quantitative analysis of defects. To overcome above challenges, we propose the DS^{2}A-Former, a novel Dual Stream Spatial Attention Transformer Network that integrates semantic and spatial information to enhance segmentation performance under complex conditions. The Dual Scale Spatial Cross Attention (DSSCA) module facilitates information exchange across scales and suppresses background noise, while the Dual Query Gated Spatial Attention (DQGSA) module uses a spatial gating mechanism to prioritize knowledge between two queries, improving representation of defect information. Extensive experiments on the MBSD dataset validate the effectiveness of our approach. We also experimented it on Crack-500 dataset, demonstrating its broad applicability in defect segmentation across domains. Our code and dataset are available at https://github.com/yikuizhai/DS2A-Former.**

***Index Terms*—Deep Learning, defect segmentation, battery surface defect, spatial information.**

This study was funded by Guangdong Basic and Applied Basic Research Foundation (No. 2021A1515011576), Guangdong Higher Education Innovation and Strengthening School Project (No. 2020ZDZX3031, No. 2022ZDZX1032, No. 2023ZDZX1029, 2024ZDZX1009), Wuyi University Hong Kong and Macao Joint Research and Development Fund (No. 2022WGALH19), Guangdong Jiangmen Science and Technology Research Project (No. 2220002000246, No. 2023760300070008390). (Corresponding author: Yikui Zhai.)

Hufei Zhu, Jie You, Yikui Zhai, Ying Xu, Tianlei Wang, Yingwen Chen are with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, 529020, China (e-mail: hufeizhu93@hotmail.com; yau_kit@163.com; yikuizhai@163.com; xuying117@163.com; tianlei.wang@aliyun.com; yingwenchen2024@163.com;).

Jianhong Zhou are with the School of Electronics and Information Engineering, South China University Of Technology, Guangzhou, Guangdong 510641, China (e-mail: jackychou_lab@126.com)

Pasquale Coscia, Angelo Genovese are with the Department of Computer Science, Università Degli Studi di Milano, 20133, Milan, Italy (e-mail: pasquale.coscia@unimi.it; angelo.genovese@unimi.it).

C. L. Philip Chen is with the Faculty of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong 510006, China (e-mail: philip.chen@ieee.org).

*Contributed to this work equally.

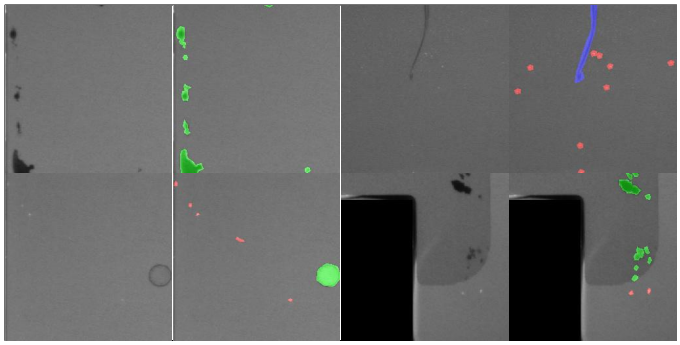

Fig. 1. Ground true labels for battery surface defect categories. Blue represents Damage, green indicates Smudge, and red denotes Dust.

I. Introduction

BATTERIES serve as essential energy carriers in mobile devices such as smartphones, tablets, and laptops. Ensuring the battery structural integrity during the manufacturing process is crucial for both safety and product quality. However, surface defects may occur due to human error or environmental factors during production. Defects such as scratches and dents can pose serious safety risks-often leading to the immediate scrapping of the battery [1], [2]. Additionally, the presence of smudge and dust can degrade the visual and perceived quality of the final product.Existing research not only lacks studies on battery surface defect data but also faces significant challenges in data collection.

In industrial quality control, the measurement and quantification of such defects by image mask, including their location, area, and type, are vital for intelligent decision-making in the field of automated manufacturing [3]. These measurements traditionally rely on Automated Optical Inspection (AOI) systems, which act as optical measurement instruments by capturing high-resolution surface images and enabling the evaluation of product quality.As shown in Fig. 1, the surface defects on batteries vary greatly in appearance, requiring advanced image-based measurement techniques for accurate assessment. However, the early proposed AOI systems and other optical measurement instruments [4] often empolied handcrafted algorithms like Canny [5] and SIFT [6], which are vulnerable to noise, lighting, and other uncontrolled envi-

TABLE I
M-BSD VS PUBLIC DEFECT DATASETS

Dataset	Num of Images			Num of Instances	Images Resolution	Available Products	Multi-categories Defects in One Image
	Train	Test	Total				
KolektorSDD [7]	347	52	399	52	500 × 1240	Electronic Commutator	No
KolektorSDD2 [8]	2331	1004	3335	356	230 × 630	Electronic Commutator	No
Magnetic Tile Dataset [9]	952	392	1344	524	512 × 512	Magnetic Tile	No
AITEX [10]	205	210	415	121	4096 × 256	Frabic	No
Crack-500 [11]	286	190	476	2974	2560 × 1440	Pavement	No
MVTec AD [12]	4960	1258	5354	1888	700 × 700 to 1024 × 1024	Daily Necessities and Materials	No
NEU-Seg [13]	3630	840	4470	13339	200 × 200	Steel Strip	Yes
Multi-class Battery Surface Defect Dataset (ours)	1983	508	2491	6636	256 × 256	Battery	Yes

ronmental factors, leading to inconsistent or inaccurate defect measurements and higher re-inspection costs.

To overcome these limitations, recent advancements in deep learning-based [14] measurement algorithms-particularly in semantic segmentation-have made it possible to obtain pixel-level measurements of surface defects, including precise boundary delineation and defect type classification. These approaches can be seamlessly integrated with visual instruments to enhance the quantitative and qualitative measurement process in automated inspection lines. Notably, [15] integrated a custom CNN method into a visual instrument to segment subtle lumber defects. Additionally, to align with the sampling rate of the visual instrument, [16] proposed a lightweight network which demonstrated excellent algorithmic efficiency in industrial defect inspection.

However, although deep learning-based methods have somewhat mitigated the impact of scene variations, significant challenges remain in fine-grained defect segmentation. Issues such as insufficient segmentation of small defects, category confusion, and inadequate pixel coverage of defect areas continue to constrain the development of defect inspection. While current methods have sufficient feature extraction capability, they still face problems of over-extraction on features and a lack of interaction between different features. This results in weak contextual relationships between features and significant loss of relevant defect information.

To address the aforementioned challenges, We construct the Multi-Class Battery Surface Defect (M-BSD) Dataset. Building upon this work, we leveraged spatial information of defects and proposed the Dual Stream Spatial Attention Transformer (DS^{2}A-Former) to enhance the model's defect pixel-level localization and recognition capabilities. First, we introduced Dual Scale Spatial Cross Attention (DSSCA) to enable spatial information interaction within deep features, resulting in multi-scale features rich in effective information. Second, during the query learning stage of DS^{2}A-Former, we incorporated Dual Query Gated Spatial Attention (DQGSA) to evaluate the effectiveness of features learned by the query. This mechanism allows the query to learn defect-related information more efficiently, thereby producing more precise predictions. The main contributions of this study can be summarized as follows:

1) The DS^{2}A-Former network was proposed, which combines Dual Scale Spatial Cross Attention (DSSCA) and the DSSA-Decoder to mitigate deep feature information loss and improve query learning. DSSCA enhances contextual relevance through spatial attention and scale fusion, while DSSA-Decoder combines cross-attention and query interaction to improve the learning process of query-based feature information, resulting in better prediction performance.
2) We proposed a query learning discrimination method that highlights useful query information before and after learning using attention mechanisms. The Dual Query Gated Spatial Attention (DQGSA) module, combined with the Spatial Distance Measurement (SDM) module, computes feature distance between the two queries. This results in the spatial boundaries of the two queries, highlighting the importance of their respective information.
3) We constructed the M-BSD dataset, which consists of 2,491 images with a resolution of 256×256, which include over 6,600 defect samples, defect samples categorized into three types: small, dense particle Dust (Dt); amorphous Smudge (Sg) without a fixed color domain; and elongated scratch-like Damage (Dm).
4) Extensive experiments were conducted on the proposed M-BSD Dataset. Our approach outperforms the baseline models, achieving improvements of 1.77% in mean Intersection over Union (mIoU), 2.89% in mean Accuracy (mAcc) and 1.95% in mean F1-score. Furthermore, it demonstrates superior performance against methods of state of art.

The remainder of this paper is organized as follows: Section II offers a brief summary of related work. Section III introduces the M-BSD Dataset. In Section IV, we present our proposed method, DS^{2}A-Former, while Section V evaluates the effectiveness of this method through extensive experiments. Finally, Section VI concludes the paper.

II. RELATED WORKS

A. Surface Defect Dataset

Recent advancements in surface defect datasets have improved defect segmentation for industrial applications. The KolektorSDD series [7], [8] includes high-resolution images of

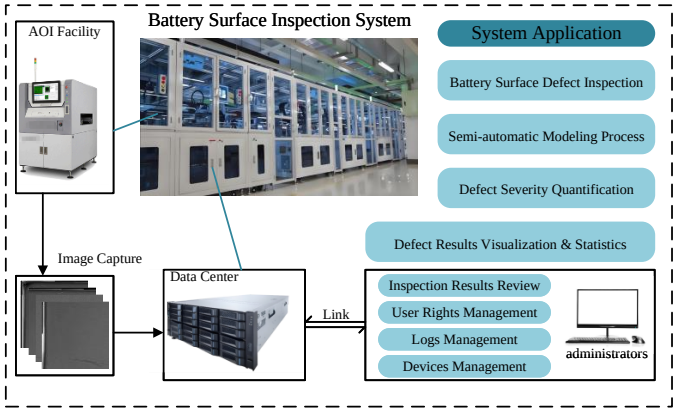

Fig. 2. Battery Surface Defect Inspection System: The components include multiple AOI cameras, several central servers, an automated production line, and remote terminals.

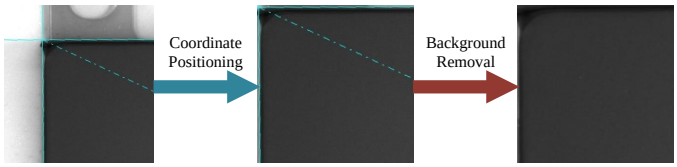

Fig. 3. Battery positioning and background removal.

metal and plastic parts, with KolektorSDD having 399 images and KolektorSDD2 containing 3,335 images, mainly focusing on cracks and scratches. The Magnetic Tile Dataset (MTD) [9] has 1,344 images with five defect types, while MVTec AD [12] includes 15 product categories and 73 defect types. The AITEX dataset [10] features 245 high-resolution images of fabrics with 12 defect types. [11] compiled a pavement crack dataset, differing from battery surface defects. However, the samples in the above datasets consist solely of defect samples from a single category per image, lacking fine-grained labeled data. Additionally, [13] expanded the NEU dataset for steel defect segmentation with 4,470 images,including three distinct defect categories, with some image samples containing defects from multiple categories.

Existing datasets often lack generalizability for battery surface defect segmentation due to variations in lighting, angles, and material properties, along with the high cost of manual defect sample collection. Furthermore, no dedicated dataset exists for finished battery surface defects. While [17] explores lithium battery defect detection, it provides limited details on datasets for finished products. To address this, we propose the M-BSD dataset, comprising 2,491 images and over 6,600 defect instances across three common defect types, specifically designed for battery surface defect segmentation tasks.

B. Surface Defect Segmentation

The primary objective of defect segmentation is to accurately identify the pixel locations of defects and classify these pixels into predefined categories. The processed pixels undergo statistical analysis and quantification, ultimately facilitating the assessment of the severity of specific defects.

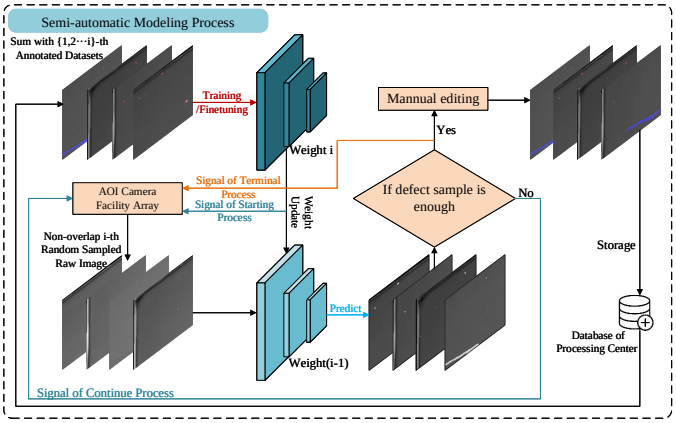

Fig. 4. Semi-Automatic Annotation Process. Step 1: Model training is performed using the i-th batch of data.Step 2: Semi-automatic annotation generates pseudo-labels, which are manually refined to form the (i+1)-th batch. This refined batch is then combined with the previous i batches for further modeling. This process can cycle through Step 1 and Step 2 iteratively.

In the early period of segmentation methods in defect inspection focused on analyzing surface textures to identify defects. Conventional, non-deep learning techniques, such as Sobel [18], Gabor [19], and GLCM [20], enhanced defect areas through spatial and frequency processing. However, these methods were limited by sensitivity to noise and lighting, requiring a stable on-site environment for effective implementation.

A decade ago, semantic segmentation became a key method for defect segmentation tasks, gaining popularity for its detailed pixel classification in industrial. Following the introduction of fully convolutional networks (FCN) [21] , convolutional neural networks advanced to pixel-level segmentation, providing classification masks for each instance. However, FCNs struggle with complex tasks, particularly in capturing small object details and losing context information during upsampling. To address above limitations, Google introduced DeepLabv1 [22], using dilated convolutions for a larger receptive field and CRF for boundary refinement, though it missed global details. DeepLabv2 [23] added ASPP for multi-scale features, increasing complexity and limiting small target accuracy. DeepLabv3 [24] refined ASPP and removed CRF but still struggled with boundary precision. DeepLabv3+ [25] combined high- and low-level features through skip connections, preserving details and handling scale changes, becoming the foundation for future segmentation models. [15] enhanced DeepLabV3+ with the GAM module for lumber defect measurement, using attention to capture long-distance dependencies and a Gaussian module to improve accuracy by reactivating blurred defect boundaries. [16] Introduced a specialized A-type architecture based on U-Net [26], designed to effectively process both low-level details and high-level semantic information of defect features.

C. Attention Mechanism

In recent years, attention mechanisms have become essential for semantic segmentation, enabling models to capture both global and local information by focusing on relevant

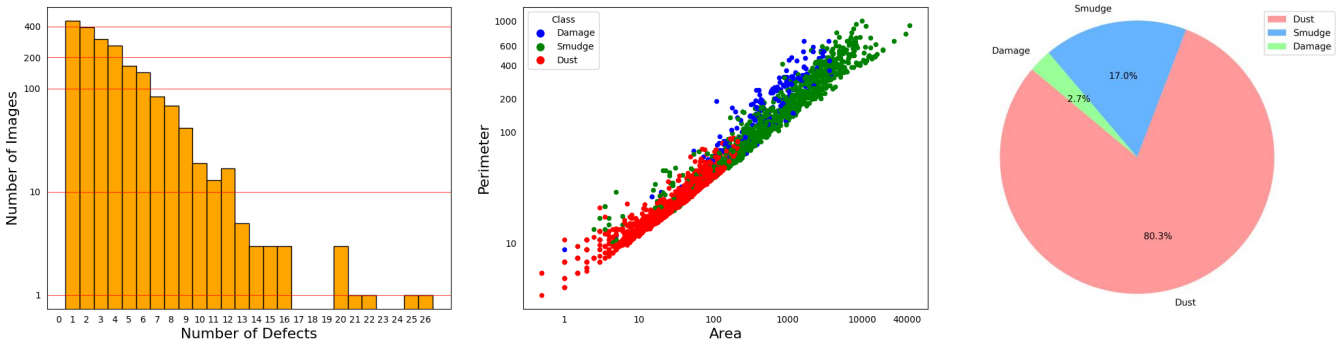

Fig. 5. Attributes of the M-BSD defect dataset. (a) Distribution of defect instances. (b) Scatter plot of area versus perimeter. (c) Proportional distribution across categories.

features and suppressing irrelevant ones. SENet [27] introduced channel-wise attention, while CBAM [28] combined channel and spatial attention for comprehensive context. Self-attention [29], cross-attention [30], and co-attention [31] mechanisms enhanced long-range feature dependencies. ViT [32] and DETR [33] introduced transformers to computer vision, establishing global pixel relationships and end-to-end object detection frameworks. MaskFormer [34] replaced pixel classification with query-based prediction, and Mask2Former [35] introduced mask and deformable attention to improve efficiency and accuracy. Additionly [36] and [37] employed self-supervised methods for pre-training to improve the generalization capability of defect segmentation. Overall, Mask2Former demonstrated broad versatility in various segmentation tasks. In our work, we adopt Mask2Former as a baseline model for the battery surface defect segmentation task.

Attention mechanisms are also essential for accurately measuring defects in the field of defect segmentation. [38] proposed a Dual Attention module for isolating small defects in capacitors. MA-Net [39] incorporates the designed SCAM module in both the encoder and decoder to enhance feature extraction and improve area restoration accuracy. This approach demonstrates strong performance in pavement crack defect inspection tasks. [40] introduced a context-aware adaptive weighted attention network with CAWAConv and gcblock, excelling on public datasets without using global max pooling. Our experiments further investigate the impact of pooling strategies.

In addition to using self-relative attention to enhance feature information, researchers have recently explored the way of interactions between different features to further extract significance information and have demonstrated the effectiveness of these approach. [41] introduced a multi-scale feature fusion method (AMFF), integrating SEAM and CEAM for fabric and aluminum defect detection, using cross-attention in FPN [42] for high accuracy and low computational cost. [43] proposed a spatial cross-attention mechanism based on RGB and sonar images, where the features of one modality are used as the query (Q) and value (V), while those of the other serve as the key (K). By applying spatial matrix operations, long-range dependencies between the two modalities are established. Although this approach lacks context perception compared to general spatial attention mechanisms, it proves effective for multimodal interaction. This insight also inspired our subsequent work, which aims to enhance context perception while facilitating interaction between the Cross-scale features to extract more effective information. [44] proposed a cross-modal feature fusion module, FAF, which applied the original image and the corresponding infrared image as input. FAF module extracts the channel attention map from one modality as a weight, applies a complement operation to it, and then performs element-wise multiplication with the other modal feature. This gated fusion method effectively leverages multi-level cross-modal information, ensuring the seamless coexistence of infrared and RGB image features, and has demonstrated strong performance in target detection. Although this method does not evaluate the distance between cross-modal features, we considered that it can be applied not only to cross-modal features but also to features in different states.

TABLE II
DEFECT INSTANCES DISTRIBUTION

Defect Categories	Num of Images			Proportion(%)	Defect Area (pixel)
	Train	Test	Total		
Damage	141	41	182	2.7	[1, 6613]
Smudge	853	273	1126	17	[5, 40000]
Dust	3759	1572	5331	80.3	[1, 312]

III. BATTERY SURFACE DEFECT DATASET

A. Battery Surface Defect Inspection System

To meet the requirements of fully automated data collection and defect inspection, we developed the Battery Surface Defect Inspection System (BSDIS), integrated into the battery production line (Fig. 2). The system consists of AOI cameras, a data processing server, and remote management terminals. Key hardware includes a conveyor belt, a status display screen (1024×768), and alarm sirens for battery transport and operator alerts.

BSDIS performs defect inspection, image acquisition, semi-automatic annotation, and system management. The AOI camera uses standardized parameters: 1.6 magnification, 1.8 depth of field, 5MP lens, and a fixed 3,000-lumen ring light. System management is via remote access, including log recording, user permissions, and role management. The data processing center handles image storage, model training, and inference, enabling defect inspection and semi-automatic annotation.

B. Image Acquisition and Preprocessing

We obtained permission from battery manufacturers to install AOI device cameras on multiple production lines. After production, a mechanical device places the batteries at specified positions on the conveyor belt every 5 seconds. The conveyor belt moves at a speed of 40mm/s, ensuring that the shutter captures battery images stably. Each AOI camera is set to capture images at 5 second intervals. To maintain a stable shooting environment, the AOI device operates with a ring light source of 3000 lumens. The conveyor belt passes through the AOI device's shooting interlayer, ensuring that battery images are captured in a semi-enclosed, stable lighting environment. Additionally, the production line's temperature and humidity are controlled at 20–24°C and 40–60% RH, respectively, to prevent material expansion. As the result, nearly 700,000 unannotated battery images were collected which were captured images at a resolution of 1600×1200.

In defect inspection, background variations across production lines can interfere with detection. To eliminate irrelevant information, we used an AOI camera's battery positioning algorithm to remove background outside the battery area (Fig. 3). Each battery image was divided into sections of 256×256 pixels, numbered, and stored as unannotated data to prevent excessive collection of defect-free images. Handling of large unannotated data volumes is discussed in Section III-C.

C. Semi-automatic Annotation Process

Battery surface defect inspection using deep learning requires diverse data for better segmentation accuracy, but obtaining sufficient samples manually is challenging. Dust particles, ranging from 3 to 200 pixels, are densely distributed with varying sizes, while Damage defects like scratches are rare, with a probability of just 0.01%. Smudges vary in size and visibilityoil stains are easily detectable, but white spots, such as metallic particles, can be misclassified. To standardize, white defects larger than 300 pixels are defined as Smudges.

To tackle these challenges, we propose a semi-automatic annotation process [45] with training and inference phases. As shown in Fig. 4, Initially, 10 samples per defect type (Dust, Smudges, Damage) are used to fine-tune pre-trained weights for segmentation. A custom filter excludes samples with defects exceeding a set limit. With sample collection growing iteratively as $10 \times (\mathrm{i}//5+1)$, where i is the iteration number.

After collecting enough samples, the model stops inference. Mask data is processed in OpenCV to extract contours, which are converted into pseudo-labels based on defect type. These labels are sent to expert annotators for refinement. Once finalized, the data is stored in the data center and merged with previous cycles to create a new dataset for training. This iterative process reduces the annotators' workload by efficiently extracting information from unannotated data.

D. Details of Multi-class Battery Surface Defect Dataset

We analyzed the M-BSD dataset to determine the total count for each category based on closed polygon masks. The results are shown in Table II, and a visualization of the dataset attributes is in Fig. 5. The dataset contains 2,491 battery partition images captured by AOI cameras and filtered through manual and model-assisted processes, with comprehensive polygon annotations for each defect type. The dataset includes 6,639 defect samples: 5,331 Dust defects, 1,126 Smudge defects, and 182 Damage defects. Each battery partition image contains an average of 2 defect samples, with a maximum of 16. The M-BSD dataset categorizes defects into three types, addressing classification challenges in battery surface defects and offering versatile samples for the lithium battery industry.

To maintain a balanced distribution of data for evaluating the segmentation model, we divided the dataset into training and testing sets using a 7:3 ratio based on defect type samples. The training set contains 72% of the original images, while the testing set, from a different time period, accounts for 28%. Both sets include diverse characteristics of three defect types: dense dust, various stains, and damages like dents and scratches. This ensures diversity in defect types, facilitating a more accurate assessment of the model's performance.

IV. Methods

A. Overall Architecture of DS^{2}A-Former

The DETR paradigm has evolved into a versatile architecture through various studies. However, while effective in natural scenes, it faces challenges in detecting battery surface defects, struggling to capture essential defect details and enable queries to learn contextual semantics from multi-scale features. These limitations stem from a core issue: the underutilization of informational disparities among multi-scale features. To address these challenges, our method combines the DETR architecture with DSSCA and DQGSA. The integrated framework of the Dual-Stream Spatial Attention Transformer (DS^{2}A-Former) is shown in Fig. 6. Next, we present the forward process of this method.

First, in the feature extraction module, the backbone network extracts features from the input image, producing shallow features at four different scales. These features are transformed into deep, multi-scale features via a pixel decoder based on Deformable Attention [46], which establishes long-range dependencies among features. Subsequently, the four deep features, ordered by decreasing resolution and semantic level from low to high, are defined as $f_d \in \{f_0, f_1, f_2, f_3\}$. These features are then input into a three-layer, dual-stream recursive feature refinement network with T-shaped dual-branch outputs, as illustrated in the gray section of Fig. 6 labeled Spatial Feature Refinement. This architecture comprises three DSSCA modules. Initially, f_3, representing the highest-level semantic feature, serving as the first query learning targets

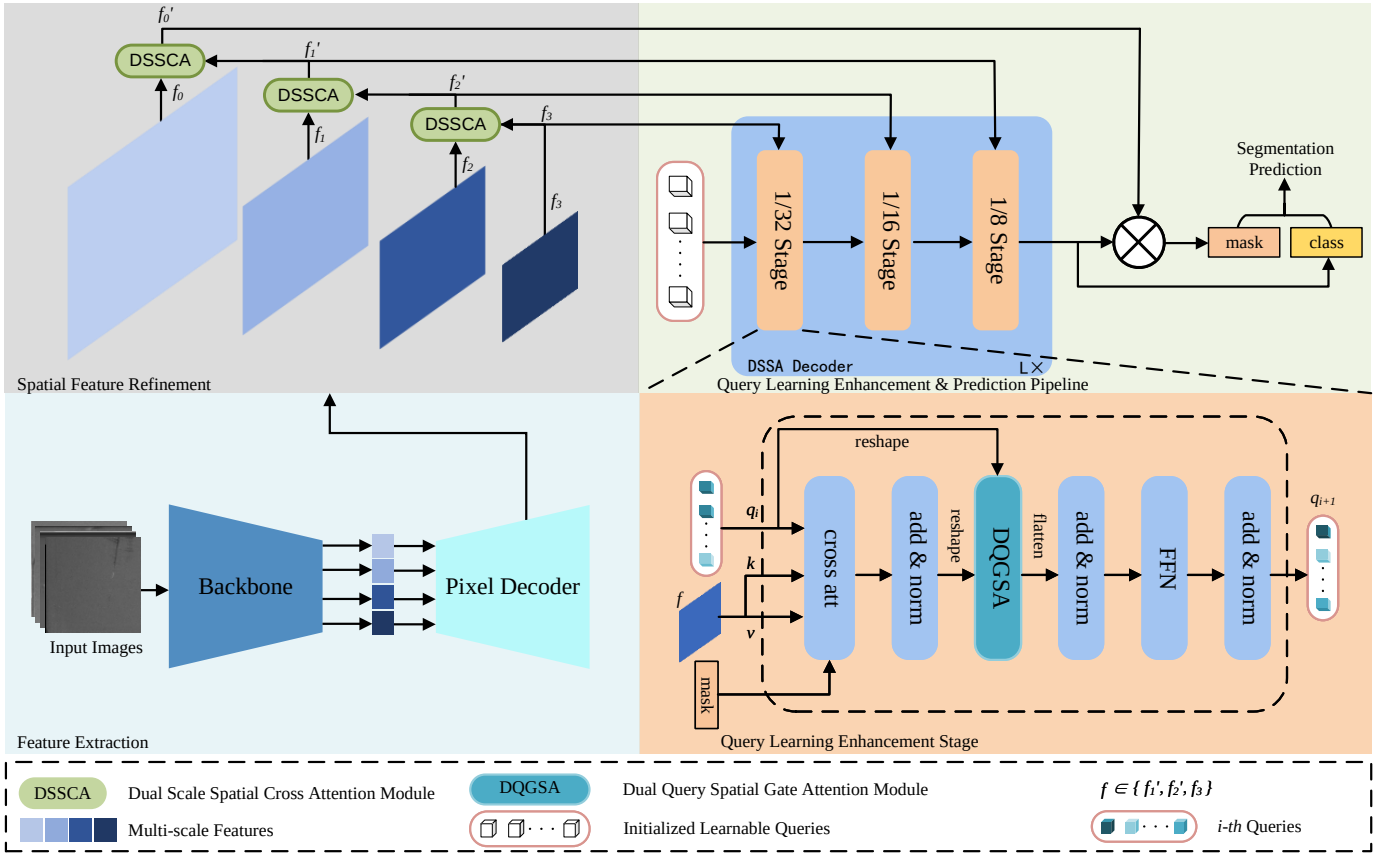

Fig. 6. Illustrates the overall framework of our proposed battery surface defect segmentation method. The DS^{2}A-Former model is built upon the DETR architecture, with core components comprising the DSSCA and DQGSA modules. Initially, DSSCA processes deep features $f_d \in \{f_0, f_1, f_2, f_3\}$, generated by the backbone and Pixel Decoder, to enable feature interactions that further enhance valuable information. These interact features are subsequently fed into the DSSA Decoder as knowledge to be learned by the initialized queries. During this process, the DQGSA module emphasizes the importance of relevant information, extracting more focused semantic and spatial details. The final output queries are then utilized to predict defect masks and classifications.

k_0 and v_0 and Simultaneously input into the first DSSCA module along with f_2 for interaction, produce f_2'. In parallel, the other branch treats f_2' as the second query learning targets k_1 and v_1, respectively. In the next stage, f_2' is used as the dual-branch output, with the left branch combining it with f_1 in a DSSCA module to yield f_1' serving as the third query learning targets k_2 and v_2. Finally, f_0' is derived similarly. Unlike previous stages, f_0' is designated as the mask feature, which undergoes matrix multiplication with the final query output. We denote DSSCA as DS, the calculation process of the dual-stream recursive architecture can be summarized as follows:

$$\begin{aligned} f_2' &= DS(f_2, f_3) \\ f_1' &= DS(f_1, f_2') = DS(f_1, DS(f_2, f_3)) \\ f_0' &= DS(f_0, f_1') = DS(f_0, DS(f_1, DS(f_2, f_3))) \end{aligned} \tag{1}$$

In the Query Learning Enhancement Stage, the query, following initialization, is input into a DSSA Decoder. A feedforward network consisting of three Query Learning Enhancement Stages. The primary design of the Query Learning Enhancement Stage is illustrated in green in Fig. 7. First, the cross-attention mechanism is introduced as a crucial approach for queries to learn image features. Like [35], we also incorporate an initialized mask to direct attention toward defect-specific regions. After the query undergoes the learning process, an addition operation followed by layer normalization is applied, and DQGSA is introduced to reshape the Bef- and Aft-learning queries into sequences with dimensions $R_q \in 10 \times 10 \times C$ for spatial information interaction. This process results in a query that contains more concentrated semantic information. Finally, query output from the Feedforward Neural Network (FFN) predicts the mask categories. An Einstein summation convention is employed with the refined mask feature to generate the new mask.

B. Dual Scale Spatial Cross Attention

To enrich contextual information and focus on defects while filtering background noise, DSSCA applies spatial attention and information compensation to adjacent-scale features before linking them with Object Queries. Traditional self-attention captures positional relationships but lacks inter-scale spatial interaction, while typical spatial attention highlights significant regions within the same scale, potentially losing cross-scale information. Inspired by [44], DSSCA enables spatial interaction across scales using dual-stream inputs and a cross-attention mechanism where low-level spatial details guide high-level semantic features. Additionally, employ original shallow semantic feature information to supplement the

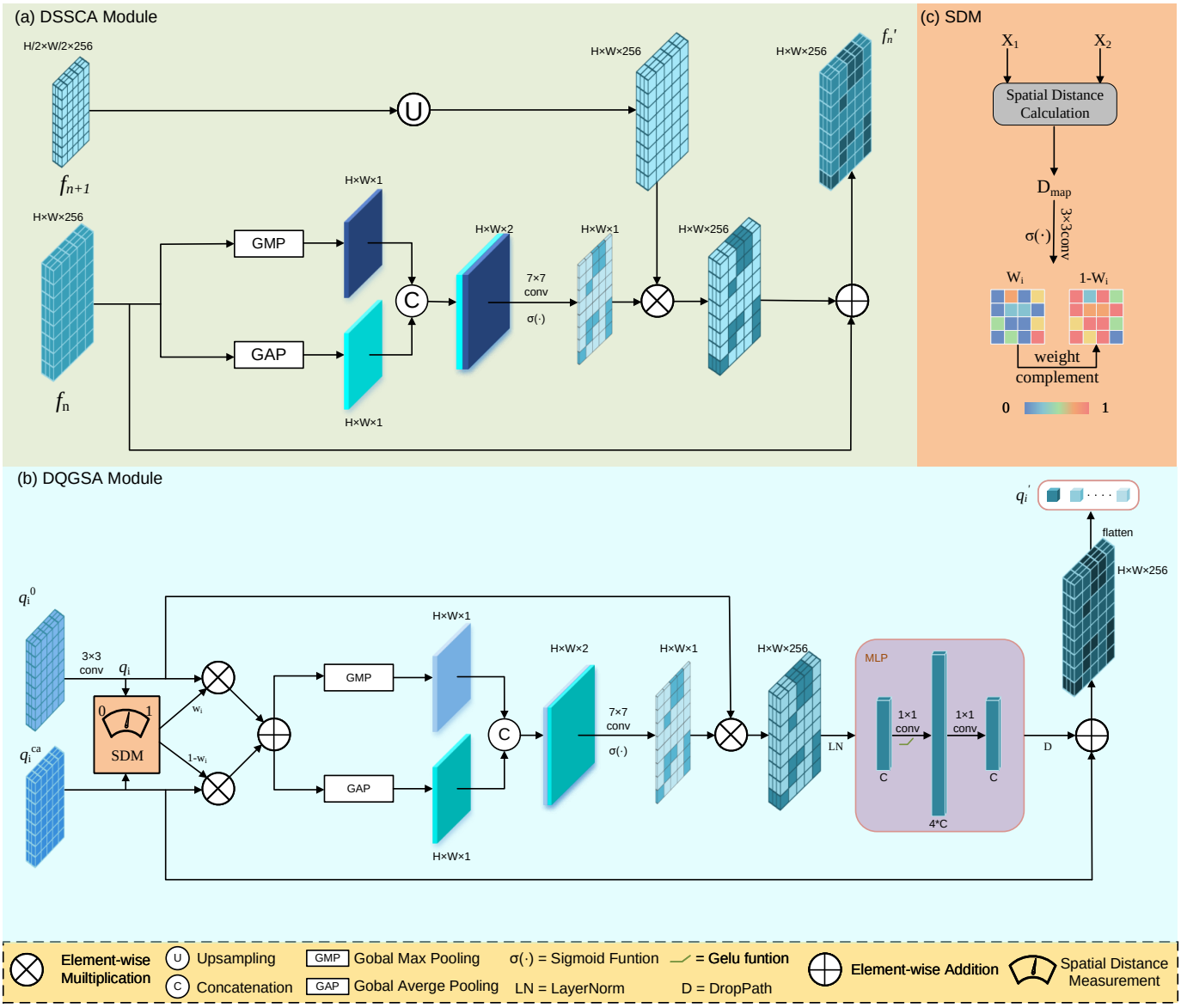

Fig. 7. (a) The DSSCA structure used for multi-scale feature interaction. (b) The structure of DQGSA for evaluating the effectiveness of query information. (c) The structure of SDM for generating gating weights.

feature map after interaction, preventing channel information loss caused by global pooling. The module design is shown in Fig. 7 (a).

After obtaining the four output scale features, we pair adjacent scale features as inputs to DSSCA, with each pair consisting of a high-level semantic feature $f_{n+1} \in \mathbb{R}^{C\times H/2\times W/2}$ and a low-level semantic feature $f_n \in \mathbb{R}^{C\times H\times W}$, where $n \in \{0, 1, 2\}$, For the low-level semantic feature f_n, spatial attention is first applied. Following the global pooling operations as in [28], channel information is compressed, and the resulting feature maps are concatenated along the channel dimension. A 7×7 convolution kernel is then used to extract useful information and suppress most irrelevant information, yielding f_n^s. Finally, a sigmoid activation function is applied to produce the normalized spatial attention map M_n.

$$f_n^s = Conv_7(Cat(GMP(f_n), GAP(f_n)) \tag{2}$$

$$M_n(i,j) = \frac{1}{1+e^{-f_n^s(i,j)}} \tag{3}$$

In this context, GMP and GAP represent global max pooling and global average pooling operations, respectively. The notation $Conv_7$ denotes a convolutional kernel of size 7×7, stride 1, padding 3, while $(i,j) \in H \times W$ represents the location of each pixel in the feature map. The term Cat refers to concatenation along the channel dimension.

For the high-level semantic feature f_{n+1}, transposed convolution is used to upsample it to match the dimensions of $f_n \in \mathbb{R}^{C\times H\times W}$, typically with an upsampling factor of 2, resulting in f'_{n+1}. Compared to bilinear interpolation, transposed convolution not only restores the resolution but also acts as a feature adjustment factor, enabling global information correction for high-level semantic features. The attention map M_n, refined from f_n acts as a strong guide for relevant regions. Given that f_{n+1} is derived from a downsampled version of

f_n, useful information may be lost in this process. By performing element-wise multiplication between f'_{n+1} and M_n, we can guide the high-level semantic information to re-weight previously neglected valuable information in f_n. Finally, the spatially refined feature f'_n is obtained by the element-wise addition of the interaction result f^n_{n+1} with the shallow feature f_n, compensating for the information lost during interaction. The complete process is outlined as follows:

$$\begin{aligned} f_n^{'} &= f_n + transconv(f_{n+1}) \times M_n \\ f_n^{'} &= f_n + f_{n+1}^{'} \times M_n \\ f_n^{'} &= f_n + f_{n+1}^{n} \end{aligned} \tag{4}$$

where $transconv$ represent a transposed convolution operation with 3×3 kernel size, stride 2, padding 1.

C. Dual Query Gated Spatial Attention

Query-based methods typically use cross-attention to learn from multi-scale features, followed by multi-head self-attention to capture dependencies among query elements. However, self-attention's high computational cost and sensitivity to background noise weaken its ability to represent sparse target features, making sparse defects harder to aware. Unlike cross-attention, which assigns weights between feature maps, self-attention distributes weights based on internal similarity, may leading to uniform attention that dilutes defect-specific focus and reduces pixel localization accuracy of defect area.

Therefore, our approach discard self-attention and posits that a single query cannot effectively capture all useful feature information during weight learning. Instead, we adopt a strategy where the query before learning, q_c, interacts with the query after cross-attention, q_{ca}, to achieve complementary information interaction between queries. As a variant of DSSCA, our DQGSA module also utilizes dual-stream features as inputs. DQGSA utilizes a spatial distance-based gating mechanism [47], [48] to allocate spatial weights between the two query features, thereby assessing the significance of information between both two queries. The resulting fused query feature, q^f, and the spatial attention map M_f derived through pooling and convolution on q^f, is used to guide q_c, preserving information in q that might have been overlooked by q_{ca} and enhancing query context-awareness. An MLP module further enhances the representational ability of the enhanced query, follow by an Element-wise addition operation by q_{ca} as information supplement. The detailed process is illustrated in Fig. 7 (b).

First, the bef-learning query $q \in \mathbb{R}^{C\times H\times W}$ is processed through a adjustment factor consisting of a 3×3 convolutional layer with stride 1 and padding 1 resulting in $q_c \in \mathbb{R}^{C\times H\times W}$. Next, both q_c and $q_{ca} \in \mathbb{R}^{C\times H\times W}$ are fed into the designed Spatial Distance Measurement (SDM) module, as shown in Fig. 7 (c). Here, H, W and C represent the feature's height, width, and number of channels, respectively. The SDM module is structured to better assess the importance of spatial information between the two features. To achieve this, the first necessary step is to compute the Euclidean distance $D_{map} \in \mathbb{R}^{H\times W}$ between q_c and q_{ca} along the channel dimension, where $c = 1, 2, 3....C$. This computation is represented by the following equation:

$$D_{map} = \| q_{ca} - q_c \|_2 = \sqrt{\sum_{c=1}^{C} \left(q_{ca(c)} - q_{c(c)}\right)^2} \tag{5}$$

where, $q_{c(c)}$ and $q_{ca(c)}$ represent the spatial information for query q_c and q_{ca} on channel c, respectively. When the values of $D_{map} \in \mathbb{R}^{H\times W}$ are smaller, it indicates closer proximity between the two features and higher similarity. Based on the values in D_{map}, a gated attention weight $W_n \in \mathbb{R}^{H\times W}$ is created for $q_c \in \mathbb{R}^{C\times H\times W}$ and $q_{ca} \in \mathbb{R}^{C\times H\times W}$ enabling weight allocation for feature importance. As shown in Fig. 7 (c), the gated attention weights $W_n = [W_1, W_2, W_3....W_{H\times W}]$ are designed to be learnable. A 3×3 convolution with stride 1 and padding 1 is applied to the spatial distance feature map D_{map} as adjustment factor, followed by a Sigmoid layer to normalize W_n. The calculation process for W_n is expressed in the following equation:

$$W_{\mathrm{n}} = \frac{1}{1 + e^{-(\mathrm{Conv}_3(D_{map}))_n}} \tag{6}$$

where, $n \in 1, 2, 3....H \times W$ represents the position of each pixel in the spatial domain, and $Conv_3$ denotes the 3×3 convolution operation. Once the attention weights are obtained, W_n and $(1 - W_n)$ are applied as control parameters for $q_c \in \mathbb{R}^{C\times H\times W}$ and $q_{ca} \in \mathbb{R}^{C\times H\times W}$, respectively. The fusion process can be describe as following equation:

$$q^f = W_n q_c + (1 - W_n) q_{ca} = {q_c}' + {q_{ca}}' \tag{7}$$

these weights adjust the fusion ratio of the two query features, resulting in ${q_c}'$ and ${q_{ca}}'$. Finally, element-wise addition is applied to merge these features, producing the fused query feature q^f.

Subsequently, spatial attention by the same manner as eq 2 and 3 is applied to extract the spatial information of q^f yielding the spatial attention map M_f. Element-wise multiplication between M_f and q_c then produces the interaction-enhanced query feature q_c^f. We utilize a layer normalization (LayerNorm) to regularize q_c^f, balancing the feature distribution. Following this, a Multi-Layer Perceptron (MLP) is utilized to refine the feature weights, with the hidden layer expanded to four times the initial node count. The MLP employs the GELU activation function and integrates DropPath regularization to enhance the stability of network convergence. Finally, an element-wise addition of q_{ca} and the refined interaction feature $q_c^{f\prime}$ is performed to compensate for any information lost due to global pooling. The primary forward process is as follows:

$$\begin{aligned} q' &= q_{ca} + MLP(Conv_3(q) \times M_f) \\ q' &= q_{ca} + MLP(q_c \times M_f) \\ q' &= q_{ca} + MLP(q_c^f) \\ q' &= q_{ca} + q_c^{f'} \end{aligned} \tag{8}$$

where the $Conv_3$ indicates a convolution layer with a 3×3 kernel size.

V. Experimental Results and Analysis

A. Experimental Setup

1) Implementation Details: ResNet-50 was used as the backbone of the network, with certain methods excluded (see Table VIII). The backbone was initialized with the pre-trained ResNet-50 [49] model from ImageNet [50]. We employed mmsegmentation as the experimental platform, and the M-BSD dataset followed the ADE20K [51] preprocessing settings, with a uniform crop size of 256×256. The initial learning rate was set to 0.0001, with a weight decay of 0.05 and a learning rate multiplier of 0.1 for the backbone. To evaluate each method's performance, we standardized iterations to 60,000 and set the batch size to 16. Data augmentation included random scaling, cropping, flipping, and color, contrast, and brightness perturbations, with a scaling factor of [0.5, 2], a cropping area ratio of 0.75, and a random flipping probability of 0.5. All experiments were conducted using the PyTorch framework on a single NVIDIA RTX 3090.

2) Evaluation Metrics: In battery surface defect segmentation, key evaluation metrics include mean Intersection over Union (mIoU), Intersection over Union (IoU), mean pixel accuracy (mAcc), and pixel accuracy (Acc). IoU measures segmentation accuracy for each defect category by calculating the ratio of the intersection to the union of predicted and ground truth areas. mIoU averages the IoU across all categories, providing an overall performance assessment. Acc indicates the proportion of correctly classified pixels, while mAcc averages pixel accuracy across categories, reflecting balanced performance. These metrics assess both the accuracy of mask coverage and pixel classification within the masks. Additionally, to provide a more comprehensive evaluation of model performance, Precision, Recall, and mean F1-score (mF1) are introduced.

$$mIoU = \Big(\sum_{i}^{N_{class}} \frac{TP_i}{TP_i + FP_i + FN_i} \Big) / N_{class} \tag{9}$$

$$mAcc = \Big(\sum_{i}^{N_{class}} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \Big) / N_{class} \tag{10}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \quad Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{11}$$

$$mF1 = \frac{1}{N} \sum_{i=1}^{N_{class}} \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{12}$$

Heres, True Positive (TP) refers to correctly identified defective pixels, while False Positive (FP) indicates incorrectly identified defective pixels. True Negative (TN) pertains to accurately identified non-defective pixels, and False Negative (FN) refers to defective pixels misclassified as non-defective. Nclass denotes the number of defect categories, including the background. mIoU measures the Intersection over Union for defective pixels, while mAcc evaluates prediction accuracy. Together, these metrics provide a comprehensive assessment of model performance. Precision indicates the accuracy of predicted defect pixels, while Recall shows how many actual defect pixels are detected. F1-score balances both, and mF1-score averages F1-scores across all classes for overall performance.

3) Baselines: We compared our proposed method with 12 representative methods, including 5 state-of-the-art and 7 classic approaches: PSPNet [52], FCN [21], DeepLabV3 [24], DeepLabV3+ [25], UPerNet [53], Segformer [54], MAE [37], BEiT [36], Mask2Former [35], Biformer [55],Vit-adapter [56] and UniRepLKNet [57]. To ensure fairness, we obtained segmentation results under identical experimental settings, except for the optimizers and initial learning rates, as variations in learning rates can lead to performance discrepancies.

B. Datasets

The M-BSD dataset provides defect samples for evaluating segmentation methods. All methods were trained on the training set, with performance assessed on the testing set. As shown in Table II, the dataset includes three defect categories: Damage (Dm) for surface scratches, Smudge (Sg) for irregular stains, and Dust (Dt) for white particles. The training set has 141 Dm, 853 Sg, and 3,759 Dt samples, while the testing set has 41, 273, and 1,672 samples, respectively. The Multi-class Battery Surface Defect (M-BSD) dataset consists of 2,491 images, each sized 256×256 pixels. We also validated the generalizability of our proposed method by applying it to the Crack-500 [11] dataset. We made no modifications to this dataset to ensure accuracy and fairness, using the training and validation sets for modeling and verifying results against the test set.

C. Ablation Experiment

In this section, We employ the Customized Mask2Former as the baseline model and conducted ablation experiments on the M-BSD dataset to validate the DS^{2}A-Former for battery surface defect segmentation. As shown in Table VIII, Mask2Former demonstrates improved performance when the self-attention layer is removed. We use this modified model as the baseline for evaluating the other proposed methods, referred to as m2f-cut-self. We emploied mIoU, mAcc and mF1 to assess validity. After confirming our method's effectiveness, we optimized performance by replacing various backbones, including ResNet18 [49], ResNet34, ResNet50, ResNet101, Swin-Transformer-Tiny [58], Swin-Transformer-Base, Swin-Transformer-Large, and ConvNeXt-Tiny [59]. For brevity, we defined R-18, R-34, R-50, R-101, Swin-T, Swin-B, Swin-L, and CNeXt-T in the table. All results represent the average of multiple trials.

1) Dual Scale Spatial Cross Attention: We conducted extensive experiments on the DSSCA module, with results summarized in Table III. The methods in the table correspond to the structural designs shown in Fig. 8. These experiments validate the effectiveness of the SAM while also examining two global pooling methods independently, as illustrated in Fig. 8 (a) and (b). Both pooling methods improved model performance, and their combined use yielded even better results, achieving mIoU, mAcc and mF1 of 55.86%, 63.24% and 68.65%, respectively. However, the CBAM, which integrates spatial and channel attention, performed significantly

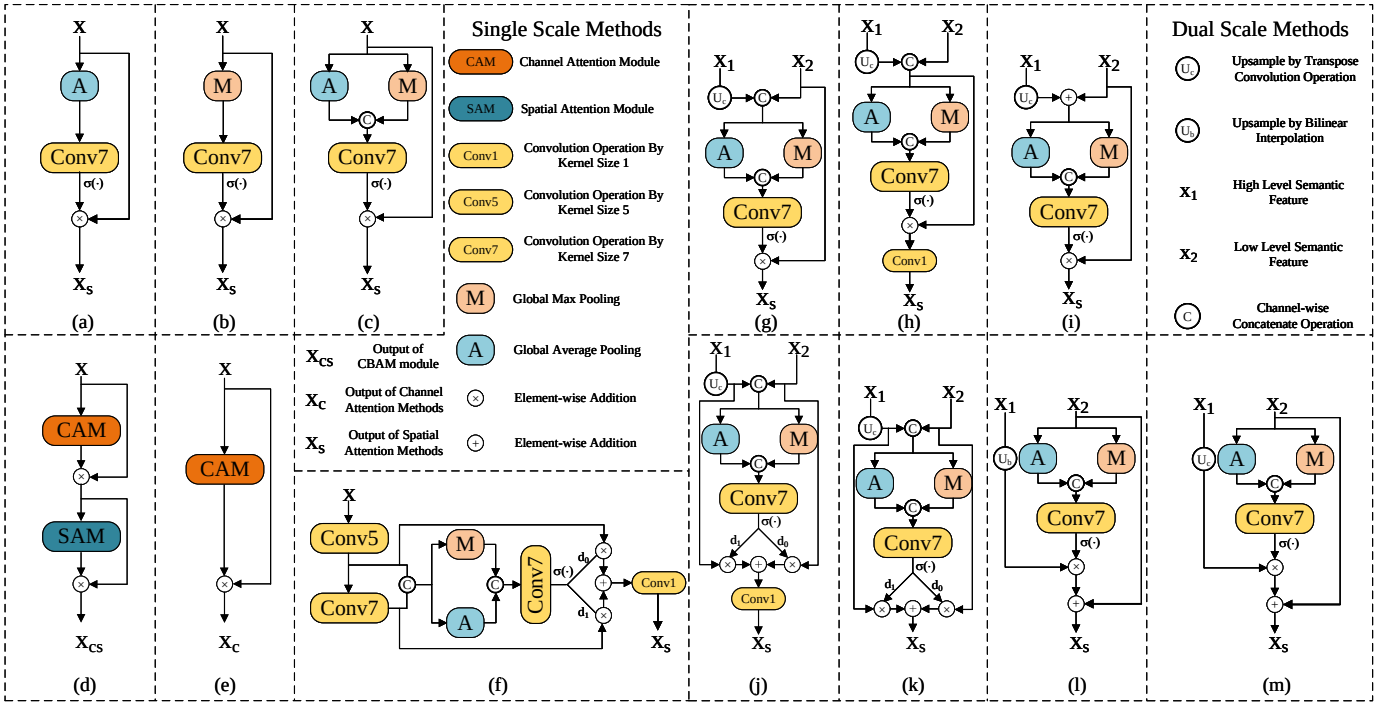

Fig. 8. The Single-stream spatial attention in the following sequence: (a) SP_{avg} module, (b) SP_{max} module, (c) spatial attention module(SAM), (d) CBAM module, (e) CAM module, (f) LSK module. The Dual-stream spatial attention is presented in the following sequence: (g) $DSSA_{cat}$ module, (h) $DSSA_{mix}$ module, (i) $DSSA_{add}$ module, (j) $DSSA_{l}$ module, (k) $DSSA_{l2}$ module, (l) $DSSCA_{b}$ module, (m) Proposed DSSCA module.

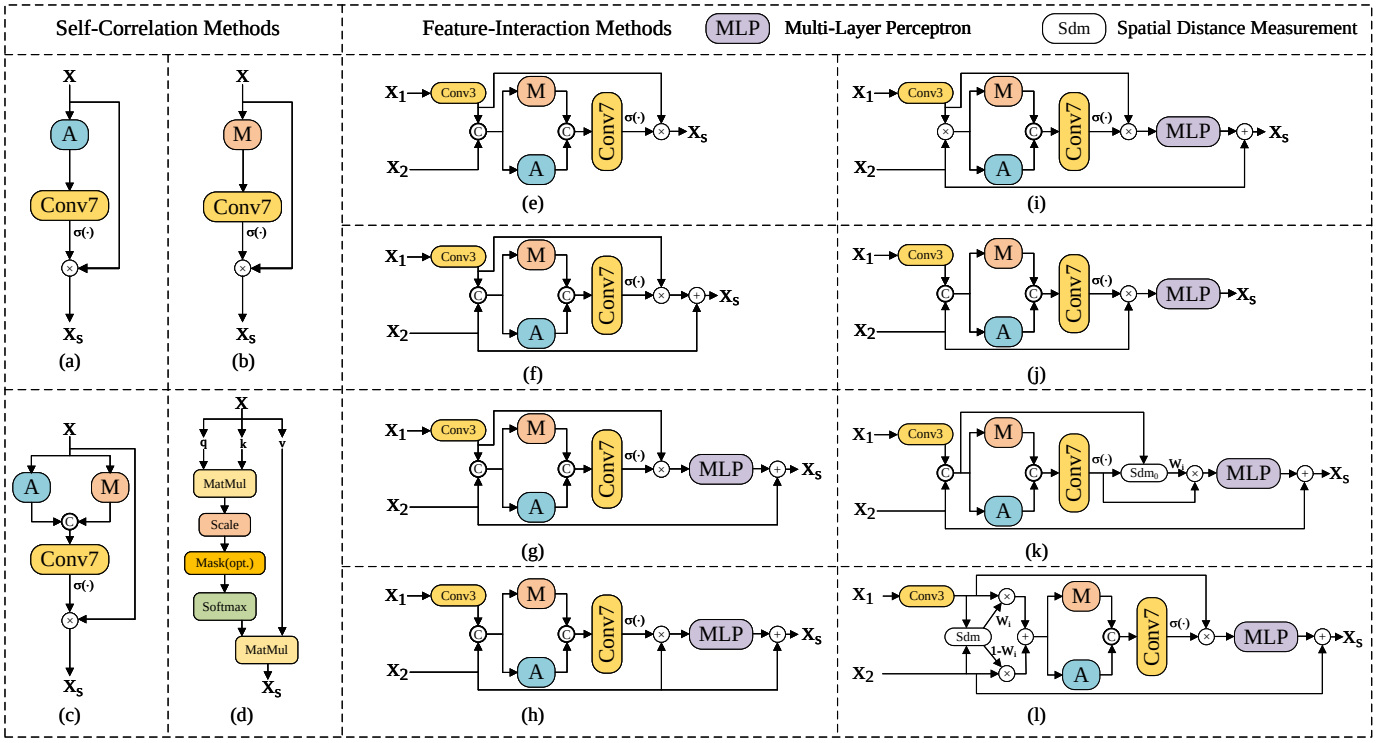

Fig. 9. Different single-stream spatial attention is displayed in the following order: (a)SP_{avg} module, (b) SP_{max} module, (c) spatial attention module(SAM), (d) self attention module. Different Dual-stream spatial attention is displayed in the following order: (e) DQSA module, (f) $DQSA_{add}$ module, (g) $DQSA_{add\text{-}M}$ module, (h) $DQSA_{x_2\text{-}add\text{-}M}$ module, (i) $DQSA_{multi\text{-}M}$ module, (j) $DQSA_{x_2\text{-}M}$ module, (k) DQSDA module, (l) Proposed DQGSA module

worse than the SAM. When using only CAM, the model failed to learn any significance information. Experimental results demonstrate that global pooling of spatial attention in channels can lead to a loss of contextual information, hindering the learning of defect features. Similarly, the LSK method, which further extracts features, caused additional loss

of deep feature information, resulting in mIoU, mAcc and mF1 of 54.62%, 60.51% and 67.38%, respectively. Nevertheless, LSK's use of multi-scale features from different downsampling stages and its proposed spatial selection mechanism are worth considering. The results highlight that while self-attention methods can refine features further, they risk losing useful information. To address this issue, we introduced the interaction of homologous neighboring features to mitigate information loss and designed the dual-scale spatial attention (DSSA) verification module. In this module, input feature resolutions range from low to high as x_1 and x_2, as shown in Fig. 8 (g) to (k). This architecture first upsamples x_1 to match the size of x_2, then fuses the two features either by concatenation or element wise addition. The fused features are used to generate the spatial attention map M_f, which is subsequently multiplied by x_2 to produce the final output. Based on different feature fusion strategies, we define DSSA_{cat} and DSSA_{add}. Additionally, depending on the target of the multiplication operation with M_f, we define DSSA_{mix}, as illustrated in Fig. 8 (g), (h), and (i). As shown in the Table III, none of these fusion strategies achieved satisfactory results, and generating the attention map after fusing x_1 and x_2 resulted in even poorer performance. Adopting spatial feature selection approach of LSK, we designed DSSA_{l} and DSSA_{l2}. The latter omits the 1×1 convolution used for output adjustment. Results show that DSSA_{l2} achieved 64.09% in mAcc; however, the critical mIoU and mF1 metric were only 55.75% and 68.57%, 0.11% and 0.08% lower than the single-stream SAM method. We redesigned DSSCA by using the cross-attention method, as shown in Fig. 8 (m). In the redesigned DSSCA, we extract the spatial attention map from x_2, which contains more feature information than high-level semantic feature x_1. This attention map represents the effective information weight of x_2. The attention weight is then element-wise multiplied with the high-level semantic feature x_1, allowing the effective information originally overlooked by x_1 to be re-extracted. Finally, the result is concatenated with the residual of x_2 to supplement the information. Compared to the general self-relative attention method, DSSCA effectively utilizes rich information which provide from cross-scale input features. Furthermore, unlike conventional cross-scale attention methods, DSSCA does not simply add, concatenate, or perform complex matrix multiplications on two features. Instead, it leverages the spatial attention map as a guiding mechanism to facilitate efficient information interaction between the features. This approach not only avoids introducing redundant information but also accurately focuses on effective defect information. AS the result, DSSCA achieved an mIoU of 56.08%, an mF1 of 68.88%, a 0.79% and a 0.90% improvement over the baseline.

In addition, we conducted experiments to verify the effect of spatial receptive field size. Typically, in many Current works [28], [60], [61], 7×7 convolution kernel is used as the receptive field in spatial attention, as it effectively captures spatial information and establishes contextual connections. To assess whether this theroy holds in battery surface defect segmentation, we replaced the spatial receptive field convolution kernel in all spatial attention modules with a 3×3 kernel for evaluation. The results indicate that, except for a few methods, most experienced a decline in mIoU, mAcc, and mF1 after adopting the smaller receptive field. This finding demonstrates that larger convolution kernels are generally more suitable for extracting spatial information from features. Finally, we compared different upsampling methods and found that transposed convolution outperforms bilinear interpolation.

TABLE III
THE COMPARISON OF DIFFERENT SPATIAL ATTENTION METHODS

Different Spatial Attention Method	Num of Used Modules	Architecture		Backbone	Size of Spatial Receptive Field	mIoU(%)	mAcc(%)	mF1-Score(%)
		Dual Scale	Single Scale					
CBAM	4		✓	ResNet-50	7×7	53.55	58.25	66.09
CAM	4		✓	ResNet-50	-	24.25	25.00	24.62
LSK	4		✓	ResNet-50	7×7	54.62	60.51	67.38
SP_{avg}	4		✓	ResNet-50	7×7	55.52	63.11	68.24
SP_{max}	4		✓	ResNet-50	7×7	55.47	60.98	67.97
SAM	4		✓	ResNet-50	7×7	55.86	63.24	68.65
DSSA_{cat}	3	✓		ResNet-50	7×7	55.58	61.32	68.29
DSSA_{mix}	3	✓		ResNet-50	7×7	55.41	62.21	68.21
DSSA_{add}	3	✓		ResNet-50	7×7	55.68	61.88	68.31
DSSCA_{b}	3	✓		ResNet-50	7×7	54.49	63.05	67.53
DSSCA	3	✓		ResNet-50	7×7	**56.08**	63.67	**68.88**
DSSA_{l}	3	✓		ResNet-50	7×7	55.17	60.72	67.89
DSSA_{l2}	3	✓		ResNet-50	7×7	55.75	**64.09**	68.57
CBAM	4		✓	ResNet-50	3×3	54.16	59.93	66.82
LSK	4		✓	ResNet-50	3×3	55.39	62.62	68.13
SP_{avg}	4		✓	ResNet-50	3×3	55.51	62.17	68.34
SP_{max}	4		✓	ResNet-50	3×3	55.57	59.69	68.19
SAM	4		✓	ResNet-50	3×3	55.31	61.75	67.98
DSSA_{cat}	3	✓		ResNet-50	3×3	55.47	61.09	68.30
DSSA_{mix}	3	✓		ResNet-50	3×3	55.30	60.85	68.02
DSSA_{add}	3	✓		ResNet-50	3×3	55.15	62.22	67.96
DSSCA_{b}	3	✓		ResNet-50	3×3	54.52	60.22	67.33
DSSCA	3	✓		ResNet-50	3×3	55.70	61.38	68.50
DSSA_{l}	3	✓		ResNet-50	3×3	55.18	61.36	67.92
DSSA_{l2}	3	✓		ResNet-50	3×3	55.40	60.03	68.14
m2f-cut-self (baseline)	-	-	-	ResNet-50	-	55.29	60.46	67.98

2) Dual Query Gated Spatial Attention: After obtaining the aggregated interaction features from the two scales, $f \in \{f_1', f_2', f_3\}$, the initialized queries serve as containers, while the features f act as the learning targets. To enhance the query's learning capability, we designed the DQGSA module, which is based on the spatial attention mechanism shown in Fig. 9. Extensive experiments were conducted, and the results are presented in Table IV.

Similarly, we first refined the query feature information using self-correlation enhancement methods, including SAM, SP_{avg}, SP_{max}, and Self-attention (Self-att). This means that the information learned by the query is actually lost after applying the above methods. Notably, as the complexity of the methods increased, the loss of useful information became more severe, and the accuracy decreased more significantly.

Thus, we adopted a Feature-Interaction approach and designed the Dual Query Spatial Attention (DQSA) architecture to validate the effectiveness of the method. x_1 and x_2 represent the queries before and after the cross-attention learning process, respectively. The validation architecture is shown in Fig. 9 (e) to (l). Initially, we concatenate the convolutional x_1 and x_2 and generate the spatial attention map M_f by the same manner of 2 and 3. The DQSA is designed by element-wise multiplication of the attention map MF_f with x_1. However, this design brings significant negative effects and severe information loss. Building on DQSA, we further introduce x_2 as supplementary information and design DQSA_{add}, as shown in Fig. 9 (f). Although some information is lost, performance is significantly improved. Subsequently, we introduced a multilayer perceptron (MLP) to refine the features and modify the feature fusion strategy. We designed $\text{DQSA}_{\text{add-M}}$ and $\text{DQSA}_{\text{multi-M}}$, as shown in Fig. 9 (g) and (i). Additionally, we changed the interaction targets of the attention mechanism and examined the necessity of supplementing

information from x_2. Based on this analysis, we designed $\text{DQSA}_{\text{x}_2\text{-add-M}}$ and $\text{DQSA}_{\text{x}_2\text{-M}}$, corresponding to Fig. 9 (h) and (j). The experimental results show that x_2 as a supplementary information source positively impacts the model performance. Moreover, the MLP enhanced the refinement of the interaction features. When the attention map interaction object is x_1, $\text{DQSA}_{\text{add-M}}$ achieved mIoU, mAcc and mF1 values of 56.32%, 62.72% and 69.03%, respectively.

Although the aforementioned methods lead to performance improvements, the fixed fusion of features lacks a mechanism to assess the validity of the information, limiting their ability to effectively handle the rich information from the two queries. We used the Euclidean distance to compute the feature distance, generating the attention map that is normalized and converted into a weight w_i to serve as the feature momentum. This approach result in the design of sdm_0. By implement idea of self-correlation, we calculated the distance between the fused features and their attention map, leading to the design of Dual Query Spatial Distance Attention (DQSDA), as shown in Fig. 9 (k). However, this approach did not yield improved performance, as it only distinguished changes in the fused features across different states, rather than effectively handling the information brought from two input queries.

We propose the Dual Query Gated Spatial Attention (DQGSA) module, which applied queries of before and after learning (x_1, x_2) as inputs. The Spatial Distance Measurement (SDM) module calculates the spatial distance between them and normalizes it through a convolution operation to generate spatial gating weights. These weights dynamically balance the effective information between the two inputs during feature fusion. The fused feature map is then refined into a spatial attention map, which is applied to x_1 via element-wise multiplication, allowing it to directly focus on valuable information. Subsequently, an MLP filters out redundant information in x_1 while further refining defect-related features. Finally, a residual connection with x_2 is employed to supplement missing information. By measuring the feature distance between the two queries, DQGSA dynamically adjusts the gating weights. The process of determining the effectiveness of the learned query information and guiding the fused features is entirely driven by spatial information which enables the model to obtain a refined query that focuses on the most relevant defect information. The final results, as shown in Table IV, demonstrate that DQGSA achieves 56.62% mIoU, 63.65% mAcc and 69.33% mF1.

To further evaluate the effectiveness of DQGSA, we adjusted several parameters of its components. Initially, we modified the ratio of the MLP, and the results showed that a ratio of 4 yielded the best performance. This also demonstrated that, even without the MLP, the method outperforms the baseline in terms of performance. Subsequently, we adjusted the kernel sizes for two modulation factors related to q and D_{map}, with the results shown in Table IV. Although the combination of 3×3 and 5×5 kernels achieved the best mAcc, but mIoU and mF1 dropped significantly. In contrast, the combination of 3×3 and 3×3 kernels provided the optimal mIoU and mF1, while also achieving an mAcc of 63.65% and this configuration are employed in subsequent experiments.

TABLE IV
THE COMPARISON OF DIFFERENT ATTENTION METHODS FOR QUERY-LEARNING

Different Attention Method for Query	Baseline	MLP Ratio	Adjustment Factor		mIoU(%)	mAcc(%)	mF1-Score(%)
			Kernel Size for q	Kernel Size for D_{map}			
SAM	DSSCA	-	-	-	55.71	61.20	68.31
SP_{max}	DSSCA	-	-	-	55.87	62.15	68.60
SP_{avg}	DSSCA	-	-	-	55.88	61.97	68.64
Self-Attention	DSSCA	-	-	-	55.76	62.32	68.18
DQSA	DSSCA	-	3×3	-	45.24	49.13	56.69
DQSA_{add}	DSSCA	-	3×3	-	55.87	60.91	68.57
$\text{DQSA}_{\text{add-M}}$	DSSCA	4	3×3	-	56.32	62.72	69.03
$\text{DQSA}_{\text{x}_2\text{-add-M}}$	DSSCA	4	3×3	-	56.29	62.03	69.07
$\text{DQSA}_{\text{muti-M}}$	DSSCA	4	3×3	-	56.12	61.42	68.78
$\text{DQSA}_{\text{x}_2\text{-M}}$	DSSCA	4	3×3	-	56.15	62.35	68.92
DQSDA	DSSCA	4	3×3	-	56.26	63.01	69.01
DSSCA	DSSCA	4	3×3	-	43.41	45.49	54.73
DQGSA	DSSCA	-	3×3	3×3	56.11	61.41	68.74
DQGSA	DSSCA	2	3×3	3×3	56.14	61.64	68.79
DQGSA	DSSCA	4	3×3	3×3	**56.62**	63.65	**69.33**
DQGSA	DSSCA	8	3×3	3×3	55.71	63.66	68.36
DQGSA	DSSCA	4	1×1	3×3	55.89	61.37	68.40
DQGSA	DSSCA	4	5×5	3×3	55.70	63.87	68.45
DQGSA	DSSCA	4	7×7	3×3	56.11	62.85	68.77
DQGSA	DSSCA	4	3×3	1×1	56.59	63.02	69.21
DQGSA	DSSCA	4	3×3	5×5	55.62	**64.18**	68.44
DQGSA	DSSCA	4	3×3	7×7	56.46	61.72	69.19
-	m2f-cut-self	-	-	-	55.29	60.46	67.98

Finally, for obtaining the best gating weights, we evaluated different feature distance calculation methods, as shown in Table V. Due to the non-differentiability of the Manhattan distance, the performance dropped significantly. In contrast, using cosine similarity for distance-weight conversion did not result in a significant performance gain, whereas the Euclidean distance method achieved the best mIoU, mAcc and mF1.

3) Dual Stream Spatial Attention Transformer: We explored three tensor strategies for DSSCA: Dual-stream Recursive (DSR) as 1, Dual-stream Independent (DSI), and Single-stream Parallel (SSP). Since DSSCA is not design for Single-stream structures, we selected spatial attention module (SAM) as the benchmark based on the best results from the single-stream structure in Table III. The other two combination strategies are defined as follows:

$$\begin{cases} f_0' = DS(f_0, f_1) \\ f_1' = DS(f_1, f_2) \\ f_2' = DS(f_2, f_3) \end{cases} \tag{13}$$

$$\begin{cases} f_0' = SAM(f_0) \\ f_1' = SAM(f_1) \\ f_2' = SAM(f_2) \\ f_3' = SAM(f_3) \end{cases} \tag{14}$$

where, DS indicating the DSSCA module in eq 13, SAM represent the spatial attention module in eq 14. Table VI summarizes the results of five implementation methods. Experimental results indicate that the single-stream parallel architecture based on SAM fails to fully optimize the model's performance. Our findings confirm that the dual-stream recursive structure achieves optimal performance, In contrast, compare to DSR, the dual-stream independent architecture, using either one or both of the proposed attention mechanisms, both exhibits poor performance. A key issue with the dual-stream independent architecture is that, although the input features are multi-scale features from the same source, the contextual relationships

TABLE V
THE COMPARISON OF DIFFERENT DISTANCE CALCULATION METHODS IN SDM

Feature Distance Calculation Methods	Calculation Target	mIoU(%)	mAcc(%)	mF1-Score(%)
Without(Remove SDM)	-	56.04	62.95	68.85
Euclidean	Bef- and Aft-Spatial Attention queries	56.26	63.01	69.01
Cosine	Bef- and Aft-learning queries	56.24	63.20	69.02
Manhattan	Bef- and Aft-learning queries	55.78	61.78	68.43
Euclidean	Bef- and Aft-learning queries	**56.62**	**63.65**	**69.33**

TABLE VI
THE COMPARISON OF DIFFERENT STRATEGIES ON DS^{2}A-FORMER

Architecture	Methods	mIoU(%)	mAcc(%)	mF1-Score(%)
SSP	SAM	55.86	63.24	67.97
DSI	DSSCA	55.41	60.45	68.32
DSR	DSSCA	**56.08**	**63.67**	**68.88**
DSI	DSSCA+DQGSA	56.07	60.59	68.70
DSR	DSSCA+DQGSA	**56.62**	**63.65**	**69.33**

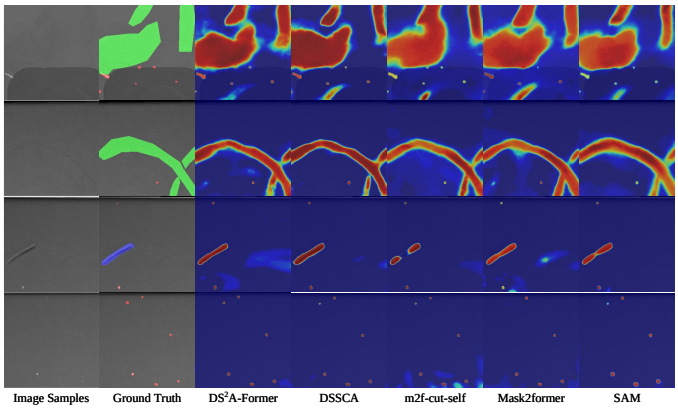

Fig. 10. Partial Heatmap of Battery Surface Defect Scores.

between the output features are much weaker compared to those output from DSR architecture.

We examined various backbones for the DS^{2}A-Former, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, Swin Transformer-Tiny, Swin Transformer-Base, Swin Transformer-Large, and ConvNeXt, to optimize performance. Results in Table VII show that Swin-B achieved the better performance, though its complexity incurs higher computational costs. ResNet-50, while slightly less effective than Swin-B, offers a balanced choice due to its comparable performance to Swin-L and fewer parameters. Both overly complex and overly simplistic backbones negatively impact performance, likely due to overfitting or underfitting. Simpler networks fail to capture essential data representations, leading to ineffective feature maps. Although deeper networks may enhance performance, excessive complexity can hinder information retention, complicating the extraction of useful information. In battery surface defect segmentation, Swin-B and ResNet-50 are optimal solutions.

4) Analysis and Discussion: To display the regions of focus for battery surface defects identified by our model, we used segmentation score maps as heatmaps for visual analysis of the DS^{2}A-Former. We employed the complete DS^{2}A-Former, the DSSCA method from Table III, a Mask2Former variant

TABLE VII
THE COMPARISON OF DIFFERENT BACKBONE UTILIZED ON DS^{2}A-FORMER

Backbone	Methods	mIoU(%)	mAcc(%)	mF1-Score(%)
R-18	DSSCA+DQGSA	53.78	59.66	66.59
R-34	DSSCA+DQGSA	55.84	61.56	68.61
R-50	DSSCA+DQGSA	56.62	63.65	69.33
Swin-T	DSSCA+DQGSA	55.50	61.33	68.13
CNeXt-T	DSSCA+DQGSA	56.21	**64.08**	69.13
Swin-B	DSSCA+DQGSA	**57.04**	63.96	**69.86**
Swin-L	DSSCA+DQGSA	56.64	63.20	69.32
R-101	DSSCA+DQGSA	56.51	62.76	69.23

without the self-attention layer (m2f-cut-self), the original Mask2Former, and a single-stream spatial attention method (SAM). The results in Fig. 10 show that removing the self-attention layer from Mask2Former slightly improves dust perception by reducing misfocused areas and concentrating on damaged regions, though it misses some critical areas. In contrast, the DSSCA method compensates for these missing regions and focuses more on all defects than SAM. However, inaccuracies remain, such as contour positions in the first row. The DS^{2}A-Former method effectively reduces attention on misfocused regions while refining focus on defect areas, improving overall model performance.

D. Overall performance

Our proposed DS^{2}A-Former architecture achieves 56.62% mIoU, 63.65% mAcc, and 69.33% mF1-score on the M-BSD dataset. The framework integrates gating mechanisms, attention mechanisms, deep semantic information, and spatial details. We compare our method with various well-known fully supervised and self-supervised techniques, with detailed results presented in Table VIII. The results demonstrate that, compared to the third best performer, Mask2Former, our approach achieves the same IoU performance in the Dm category but an Acc increase in 2.13%. In the Sg category, our method outperforms Mask2Former with improvements of 2.75% in IoU and 2.89% in Acc. For small objects in the Dt category, our method improves IoU and Acc by 4.28% and 6.47%, respectively. In terms of precision and recall, some methods, such as UperNet, exhibit an mPrecision of 86.44% but an mRecall of only 50.72%, indicating that although the predicted defect areas are accurate, many actual defect areas are missed. Our method strikes a relatively better balance between mPrecision and mRecall, suggesting that it can capture most actual defect areas with fewer false positives. Consequently, the final mF1-score is higher than that of other methods. Overall, our method improves mIoU, mAcc, and mF1 by 1.77%, 2.89%, and 1.95%, respectively. Although our approach is outperform in M-BSD, but it also exist limitation. In the Dm category, there remains a gap compared to the universal methods, Vit-adapter and UniRepLKNet.

Similarly, the experiments on the crack-500 dataset are shown in Table IX. The results show that our method is also competitive in defect inspection in different fields, reaching 72.75% mIoU, 81.93% mAcc and 81.64% mF1. Although our method achieves a lower mF1-score than MANet, Other results

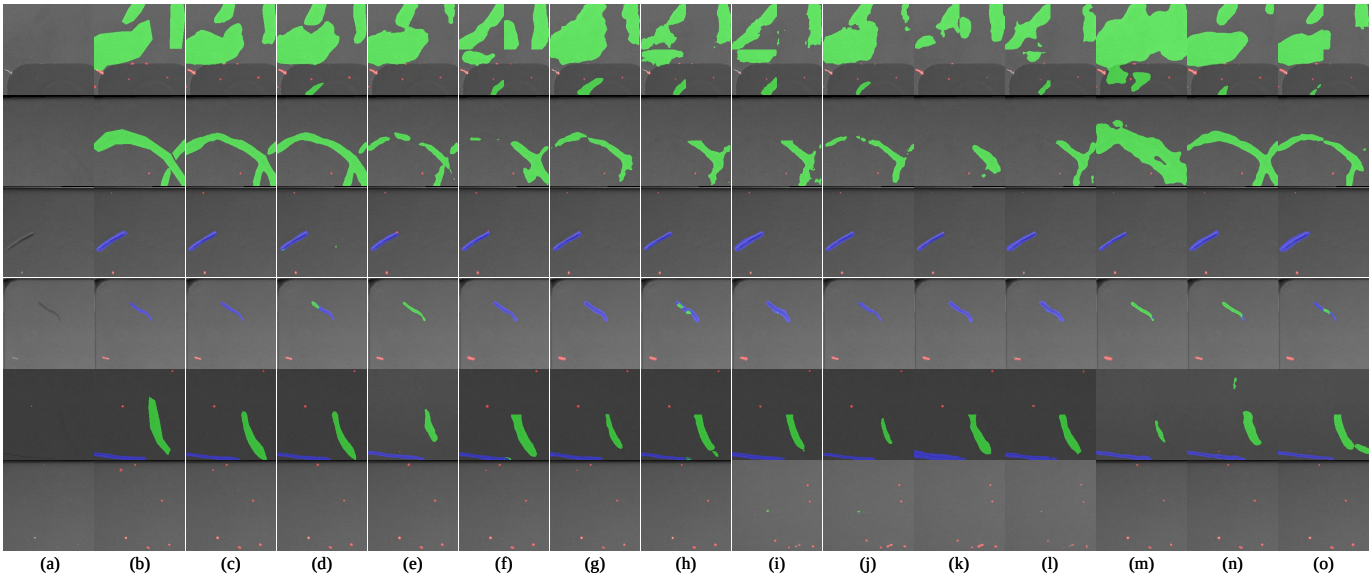

Fig. 11. Segmentation Results of battery surface defect. (a) Input image, (b) Ground truth, (c) Proposed DS^{2}A-Former, (d) Mask2former, (e) Segformer-b3, (f) MAE, (g) Upernet, (h) Deeplabv3+, (i) DeepLabv3, (j) PSPNet, (k) BEiT, (l) FCN,(m) Biformer, (n) UniRepLKNet, (o) ViT-Adapter. Blue, red and green represent Damage, Dust, Smudge respectively.

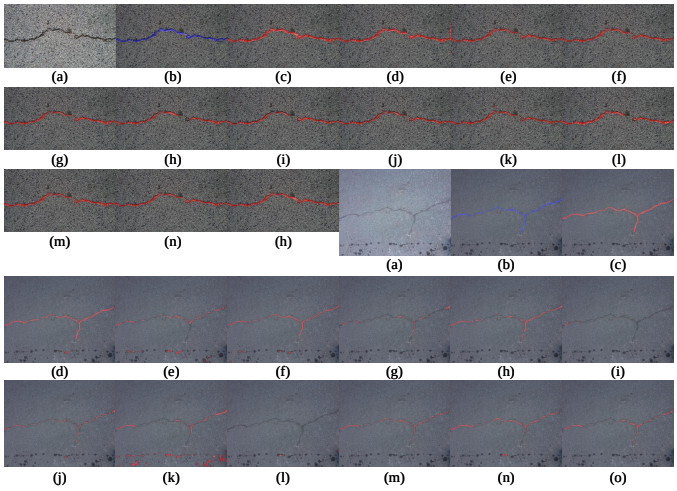

Fig. 12. Segmentation Results on Crack-500. (a) Input image, (b) Ground truth, (c) Proposed DS^{2}A-Former, (d) Mask2former, (e) Segformer-b3, (f) MAE, (g) Upernet, (h) Deeplabv3+, (i) DeepLabv3, (j) PSPNet, (k) BEiT, (l) FCN, (m) Biformer, (n) UniRepLKNet, (o) ViT-Adapter. Blue and red represent GT label and prediction result.

are 0.32% and 0.63% higher than the second best methods in mIoU and mAcc respectively.

In pervious section we presented the heatmap of defect areas that our proposed method focuses on in Fig. 10. However, the classification ability within these focused areas is also crucial and warrants further mask visualization. Fig. 11 compares segmentation results on the M-BSD dataset. Mask2Former, SegFormer, DeepLabV3+, DeepLabV3, BiFormer, UniRepLKNet, and ViT-Adapter misclassify partial Dm areas as Sg, while MAE, PSPNet, BEiT, FCN, and our proposed DS^{2}A-Former accurately identify battery damage. In the sixth row of Fig. 11, we can clearly observe that MAE, DS^{2}A-Former, ViT-Adapter, and UniRepLKNet detect all dust instances. However, in terms of mask coverage completeness, only DS^{2}A-Former produces results closest to the ground truth, while the other methods exhibit missed detections and misclassifications of irrelevant areas. In final Smudge segmentation, current methods, except Mask2Former, show significant false positives and missed detections, indicating deficient performance. The DS^{2}A-Former, however, achieves substantial coverage that closely matches the ground truth without false detections.

We also performed a visual analysis of the results on the Crack-500 dataset. As shown in Fig. 12, the DS^{2}A-Former we proposed demonstrates strong performance in road crack inspection. To evaluate the model's capabilities, we tested cracks at two granularities. The results show that, in fine-grained crack segmentation, all methods except ours produce incoherent masks, while our method successfully segments a complete crack mask. Similarly, at the coarse granularity, our proposed method closely aligns with the ground truth, while other methods, except for Mask2Former (which is the closest to the ground truth), suffer from significant mask coverage deficiencies. Additionally, some methods incorrectly classify irrelevant areas as cracks, with BEiT being the most prone to this problem.

VI. Conclusion

This study addresses the lack of battery surface defect data in the defect inspection industry by creating the M-BSD Dataset using AOI cameras. We propose the DS^{2}A-Former architecture to improve defect pixels localization and recognition, considering both the similarities within defect categories and the differences between them, as well as the high similarity between defects and backgrounds. The DSSCA module integrates spatial information across scales, enhancing semantic context and reducing background interference. Additionally, the DQGSA module adaptively emphasizes the importance of Bef- and Aft-learning information of queries,

TABLE VIII
SEGMENTATION RESULTS OF DIFFERENT METHODS FOR M-BSD DATASET

Methods	Backbone	Sg(%)				Dm(%)				Dt(%)				mIoU(%)	mAcc(%)	mF1-Score(%)	mPrecision(%)	mRecall(%)
		IoU	Acc	Precision	Recall	IoU	Acc	Precision	Recall	IoU	Acc	Precision	Recall					
FCN [21]	ResNet-50	24.39	28.68	61.98	28.68	45.78	51.83	79.68	51.83	12.96	13.72	69.85	13.72	45.16	48.44	55.92	77.37	48.44
Deeplabv3 [24]	ResNet-50	24.25	28.05	64.17	28.05	49.71	59.61	74.96	59.61	24.87	26.98	76.05	26.98	49.11	53.55	61.01	78.30	53.55
Deeplabv3+	ResNet-50	25.51	29.75	64.20	29.75	47.54	54.88	78.05	54.89	25.93	27.79	79.48	27.79	49.15	53.00	61.27	79.94	53.00
Upernet [53]	ResNet-50	24.72	26.36	79.92	26.36	52.91	59.53	82.64	59.53	26.73	28.50	81.11	28.50	50.54	53.55	62.48	85.41	53.55
Segformer-B3 [54]	MV-Former	27.42	29.96	76.40	29.96	54.58	61.03	83.78	61.03	33.32	39.23	68.88	39.23	53.29	57.49	65.64	81.79	57.49
MAE [37]	MAE	30.93	38.21	61.87	38.21	50.1	55.27	84.27	55.27	30.57	33.62	77.08	33.62	52.32	56.63	64.91	80.37	56.63
BEiT [36]	BEiT	19.82	22.98	59.03	22.98	42.91	54.74	66.51	54.74	27.82	30.21	77.83	30.21	47.00	51.87	58.84	75.31	51.87
PSPNet [52]	ResNet-50	18.84	19.59	**83.23**	19.59	49.47	53.49	86.81	53.49	27.59	29.93	77.92	29.93	48.40	50.72	59.99	**86.44**	50.72
Mask2former [35]	ResNet-50	32.7	46.22	52.78	46.22	56.52	61.37	**87.75**	61.37	32.72	36.50	75.92	36.50	54.85	60.76	67.38	78.73	60.76
Biformer-Base [55]	Biformer-B	24.60	32.42	50.50	32.42	56.14	61.03	87.50	61.03	28.31	30.84	77.54	30.84	51.59	55.87	63.54	78.41	55.87
UniRepLKNet-Base [57]	UniRepLKNet-B	29.78	32.76	76.57	32.76	58.76	**70.75**	77.61	**70.75**	34.49	38.74	75.84	38.74	55.23	60.49	67.53	82.05	60.49
ViT-Adapter-Base [56]	ViT-Adapter-B	27.82	30.35	76.94	30.35	**59.68**	68.45	82.33	68.45	30.57	33.25	79.13	33.25	53.98	57.94	66.00	84.12	57.94
m2f-cut-self (baseline)	ResNet-50	35.03	43.17	65.02	43.16	54.28	61.32	82.52	61.32	34.05	37.94	76.85	37.94	55.29	60.46	67.98	80.70	60.45
DS^{2}A-former(ours)	ResNet-50	**35.45**	**49.11**	56.04	**49.11**	56.46	63.50	83.59	63.50	**37.00**	**42.97**	72.71	**42.97**	**56.62**	**63.65**	**69.33**	77.72	**63.65**

TABLE IX
SEGMENTATION RESULTS OF DIFFERENT METHODS FOR THE CRACK-500-TEST DATASET

Methods	Backbone	mIoU(%)	mAcc(%)	mF1-Score(%)
FCN [21]	ResNet-50	62.67	65.80	70.96
PSPNet [52]	ResNet-50	58.69	69.17	73.95
DeepLabv3 [24]	ResNet-50	61.72	63.8	69.74
DeepLabv3+ [25]	ResNet-50	68.08	71.93	77.03
BEiT [36]	BEiT	71.74	79.96	80.70
UPerNet [53]	ResNet-50	70.73	76.43	79.71
Segformer-B3 [54]	MV-Former	70.63	75.63	79.60
MAE [37]	MAE	72.43	78.53	81.31
Mask2former [35]	ResNet-50	72.41	81.36	81.33
MANet [39]	MDSC-MobileNet	72.19	-	**85.77**
BiFormer-Base [55]	BiFormer-B	70.51	75.52	79.48
UniRepLKNet-Base [57]	UniRepLKNet-B	70.89	76.10	79.85
ViT-Adapter-Base [56]	ViT-Adapter-B	71.79	78.48	80.72
DS^{2}A-former (ours)	ResNet-50	**72.75**	**81.93**	81.64

using Euclidean distance for refined feature extraction. We validate the DS^{2}A-Former framework and its modules through extensive experiments, achieving state-of-the-art performance on both the M-BSD and Crack-500 datasets. However, we also observed that the Euclidean distance calculation method can result in excessively large numerical values during the feature calculation process, which may lead to model instability.

In the future, our goal is to further expand the battery surface defect dataset, explore advanced feature interaction methods for representation learning, optimize feature distance calculation methods, maximize the effective utilization of feature information, simplify the model structure, and enable the model to adapt its interaction methods based on feature depth.

REFERENCES

[1] F. Baronti, C. Bernardeschi, L. Cassano, A. Domenici, R. Roncella, and R. Saletti, "Design and safety verification of a distributed charge equalizer for modular li-ion batteries," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1003–1011, 2014.

[2] D. Li, J. Deng, Z. Zhang, Z. Wang, L. Zhou, and P. Liu, "Battery safety risk assessment in real-world electric vehicles based on abnormal internal resistance using proposed robust estimation method and hybrid neural networks," *IEEE Transactions on Power Electronics*, vol. 38, no. 6, pp. 7661–7673, 2023.

[3] Y. Qin, C. Yuen, X. Yin, and B. Huang, "A transferable multistage model with cycling discrepancy learning for lithium-ion battery state of health estimation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1933–1946, 2022.

[4] G. C. Giakos, L. Fraiwan, N. Patnekar, S. Sumrain, G. B. Mertzios, and S. Periyathamby, "A sensitive optical polarimetric imaging technique for surface defects detection of aircraft turbine engines," *IEEE Transactions on instrumentation and measurement*, vol. 53, no. 1, pp. 216–222, 2004.

[5] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[6] A. H. Madessa, J. Dong, E. Rigall, Q. Lv, H. S. N. Nawaz, I. Mugunga, and S. Guo, "Transmittance surface detection and material identification using multitask vit-sift fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–17, 2022.

[7] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-Based Deep-Learning Approach for Surface-Defect Detection," *Journal of Intelligent Manufacturing*, May 2019.

[8] J. Božič, D. Tabernik, and D. Skočaj, "Mixed supervision for surface-defect detection: from weakly to fully supervised learning," *Computers in Industry*, 2021.

[9] Y. Huang, C. Qiu, and K. Yuan, "Surface defect saliency of magnetic tile," *The Visual Computer*, vol. 36, no. 1, pp. 85–96, 2020.

[10] J. Silvestre-Blanes, T. Albero-Albero, I. Miralles, R. Pérez-Llorens, and J. Moreno, "A public fabric database for defect detection methods and results," *Autex Research Journal*, vol. 19, no. 4, pp. 363–374, 2019.

[11] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2019.

[12] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.

[13] 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 Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] Y. Zhong, Z. Ling, L. Liu, S. Zhang, and H. Wen, "Deep gaussian attention network for lumber surface defect segmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.

[16] B. Chen, T. Niu, W. Yu, R. Zhang, Z. Wang, and B. Li, "A-net: An a-shape lightweight neural network for real-time surface defect segmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–14, 2024.

[17] K. Wu, J. Tan, and C. Liu, "Cross-domain few-shot learning approach for lithium-ion battery surface defects classification using an improved siamese network," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 11 847–11 856, 2022.

[18] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

[19] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[20] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[23] ——, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[24] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
[25] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.
[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.
[27] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
[28] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 6000–6010.
[30] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *International Conference on Machine Learning (ICML)*, 2021, pp. 4651–4664.
[31] J. Li, H. Qi, B. Dai, X. Ji, and L. Carin, "Visual grounding via dual-path modeling for text-image attention," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 10 939–10 949.
[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2020.
[33] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.
[34] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
[35] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1290–1299.
[36] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.
[37] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021.
[38] J. He, X. Cao, K. Takamasu, and M. Chen, "Machine vision-based lightweight multi-scale automatic defect segmentation network for mlcc," *IEEE Transactions on Instrumentation and Measurement*, 2024.
[39] J. Chen, Y. Wen, Y. A. Nanehkaran, D. Zhang, and A. Zeb, "Multiscale attention networks for pavement defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.
[40] G. Zhang, Y. Lu, X. Jiang, F. Yan, and M. Xu, "Context-aware adaptive weighted attention network for real-time surface defect segmentation," *IEEE Transactions on Instrumentation and Measurement*, 2024.
[41] Y. Zhao, Q. Liu, H. Su, J. Zhang, H. Ma, W. Zou, and S. Liu, "Attention-based multi-scale feature fusion for efficient surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, 2024.
[42] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
[43] Y. Li, B. Wang, J. Sun, X. Wu, and Y. Li, "Rgb-sonar tracking benchmark and spatial cross-attention transformer tracker," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 35, no. 3, pp. 2260–2275, 2025.
[44] X. Li, Y. Li, H. Chen, Y. Peng, and P. Pan, "Ccafusion: Cross-modal coordinate attention network for infrared and visible image fusion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 2, pp. 866–881, 2024.
[45] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert *et al.*, "Incremental classifier and representation learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 5533–5542.
[46] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable {detr}: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=gZ9hCDWe6ke
[47] A. Graves and A. Graves, "Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.
[48] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.
[51] Y. Zhou and E. Tartaglione, "Semantic understanding of scenes through the ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
[52] H. Zhao, J. Shi, X. Qi, S. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6230–6239.
[53] T. Xiao, Z. Liu, and S. Zhang, "Unified perceptual parsing for scene understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 10–12.
[54] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in neural information processing systems*, vol. 34, pp. 12 077–12 090, 2021.
[55] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. Lau, "Biformer: Vision transformer with bi-level routing attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 10 323–10 333.
[56] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," *arXiv preprint arXiv:2205.08534*, 2022.
[57] X. Ding, Y. Zhang, Y. Ge, S. Zhao, L. Song, X. Yue, and Y. Shan, "Unireplknet: A universal perception large-kernel convnet for audio video point cloud time-series and image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 5513–5524.
[58] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, B. Li, and D. Yu, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.
[59] Z. Liu, J. Wu, C. Xu, and et al., "Convnext: Revisiting convolutional neural networks with transformers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
[60] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," *Computational visual media*, vol. 9, no. 4, pp. 733–752, 2023.
[61] Z. Ying, J. Zhou, Y. Zhai, H. Quan, W. Li, A. Genovese, V. Piuri, and F. Scotti, "Large-scale high-altitude uav-based vehicle detection via pyramid dual pooling attention path aggregation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 10, pp. 14 426–14 444, 2024.

Hufei Zhu received academic training in Electrical Engineering (B.S.E) from Beijing University of Posts and Telecommunications, Signal Processing (M.S.) from the South China University of technology, and Safety engineering (Ph.D) from Beijing Jiaotong University.

His research interests lie at the intelligent systems, control theory and pattern recognition.

Jie You received the B.S. degree from Wuyi University, Jiangmen, China, in 2019. He is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University.

His research interests include: computer vision, defect segmentation, and image processing.

Yikui Zhai (Senior Member, IEEE) received his Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in June 2013. Since October 2007, he has been working with Wuyi University, Jiangmen, China, where he is a Full Professor now. He is also Associate Dean of the School of Electronics and Information Engineering in Wuyi University, since 2024. He has been a Visiting Scholar with Department of Computer Science, the Universita degli Studi di Milano, Italy, during June 2016 to June 2017, August 2023 and January 2024. His research interests include: Image Processing, Deep Learning, Optical Character Recognition, Object Detection, UAV Change Detection, Self Supervise Learning.

Ying Xu received the BS. and MS. degrees in automation, and control theory and control engineering from the Wuhan University of Science and Technology, Wuhan, China, in 2004 and 2008, respectively, and the Ph.D. degree in control theory and control engineering from the South China University of Science and Engineering, Guangzhou, China, in 2013. She is currently an Associate Professor with the School of information and communication engineering, Wuyi University, Jiangmen, Guangdong, China. Her current research interests include image processing, deep learning, and biometric extraction.

Tianlei Wang received academic training in Electrical Engineering (B.S.E) from Beijing University of Posts and Telecommunications, Signal Processing (M.S.) from the South China University of technology, and Safety engineering (Ph.D) from Beijing Jiaotong University. His research interests lie at the intelligent systems, control theory and pattern recognition.

Yingwen Chen received the B.S.degree in Big Data Management and Application from Neusoft Institute Guangdong in 2023. He is currently working toward the M.S. degree degree with the School of Electronics and Information Engineering, Wuyi University.

His research interests include computer vision, pattern recognition, and image processing.

Jianhong Zhou (Member, IEEE) is currently pursuing the Ph.D. degree in the School of Electronics and Information Engineering, South China University of Technology. He received the B.S. and MS. degree in Wuyi University of Communication Engineering and Information and Communication Engineering in 2020 and 2024. His research interests include computer vision, representational contrastive learning, image processing and object detection.

Pasquale Coscia (Senior Member, IEEE) received the Ph.D. degree in Industrial and Information Engineering from the Universita degli Studi della Campania Luigi Vanvitelli, Italy, in 2019.

From 2019 to 2022, he was a post-doctoral researcher at the Universita degli Studi di Padova, Italy. He is currently an Assistant Professor at the Department of Computer Science of the Universita degli Studi di Milano, Italy. He is Co-chair of the Intelligent Measurement Systems Technical Committee (TC-22) of the IEEE Instrumentation and Measurement Society since 2023. His research activities are focused on theoretical, methodological, and applied aspects of computational intelligence for signal and image processing.

Angelo Genovese (Senior Member, IEEE) received the Ph.D. degree in computer science from the Universita degli Studi di Milan, Crema, Italy, in 2014. He has been a postdoctoral Research Fellow in computer science with the Universita degli Studi di Milan since 2014.

He has been a Visiting Researcher with the University of Toronto, Toronto, ON, Canada. Original results have been published in over 30 papers in international journals, proceedings of international conferences, books, and book chapters.

His current research interests include signal and image processing, three-dimensional reconstruction, computational intelligence technologies for biometric systems, industrial and environmental monitoring systems, and design methodologies and algorithms for self-adapting systems.

Dr. Genovese is an Associate Editor for the Journal of Ambient Intelligence and Humanized Computing (Springer).

C. L. Philip Chen (Fellow, IEEE) received the M.S. degree in electrical engineering from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He was a tenured Professor, the Department Head, and an Associate Dean of two different universities in the U.S. for 23 years. He is currently the Head of the School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong, China. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen received the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University. He has been the Editor-in-Chief of the IEEE TRANSACTION ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS since 2014 and an associate editor of several IEEE TRANSACTIONS. He was the Chair of TC 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017 and also a Program Evaluator of the Accreditation Board of Engineering and Technology Education of the U.S. for computer engineering, electrical engineering, and software engineering programs. He was the IEEE SMC Society President from 2012 to 2013 and a Vice President of Chinese Association of Automation (CAA). He is a Fellow of AAAS, IAPR, CAA, and HKIE.