Background-Adaptive Surface Defect Detection Neural Networks via Positive Samples

1st Tongzhi Niu*
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
tzniu@hust.edu.cn

2nd Biao Chen*
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
u202010899@hust.edu.cn

3th Zhenrong Wang
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
zora_wang@hust.edu.cn

4rd Ruoqi Zhang
China-EU Institute for Clean and Renewable Energy
Huazhong University of Science and Technology
Wuhan, China
m202271390@hust.edu.cn

5rd Bin Li**
School of Mechanical Science and Engineering
Huazhong University of Science and Technology
Wuhan, China
libin999@hust.edu.cn

***Abstract*—In this paper, we address the challenge of surface defect detection in manufacturing, particularly under conditions of background variation and noise interference. To tackle this issue, we propose a novel Background-Adaptive Surface Defect Detection Network (BANet). The proposed BANet enhances the defect detection capabilities by improving generalization capacity through learning comparative abilities between positive samples and testing samples. In order to mitigate the impact of three types of noise (texture variation, translation, and rotation), we introduce a Foreground Edge Attention Mechanism (FEAM) and a Spatial Transformer Module (STM). The FEAM enhances the model's ability to differentiate between foreground and background, thereby effectively reducing texture variation noise. The STM uses affine transformations to eliminate translation and rotation noise. Our proposed network was validated on two - Optical Communication Devices (OCDs) dataset, and demonstrated superior performance over the state-of-the-art methods. The findings of this study highlight the potential of our approach in effectively addressing surface defect detection in variable backgrounds and noisy conditions, thereby contributing significantly to the quality and reliability of manufacturing processes.**

***Index Terms*—Surface Defect Detection, Background-Adaptive, Positive Sample based, Spatial Transformer Networks**

I. Introduction

In the rapidly accelerating realm of industrialization and advanced manufacturing, the importance of defect detection has surged dramatically. Playing a pivotal role across numerous sectors, such as materials science, manufacturing, aviation, electronics, and quality control, it underpins product quality, reinforces process reliability, and reduces wastage. Furthermore, it contributes to ensuring the structural integrity and functionality of the end-products, thereby directly affecting customer safety and satisfaction.

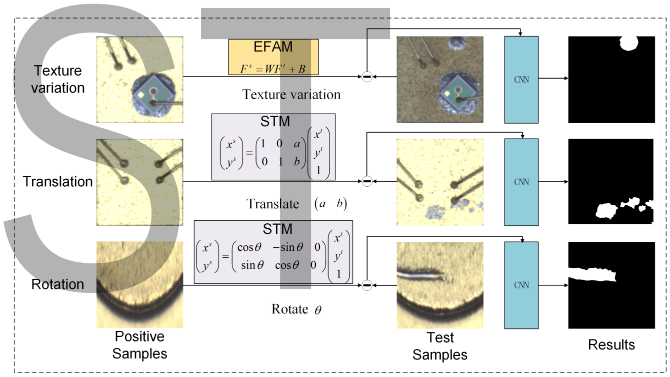
Texture variation
Translation
Rotation
EFAM
$F^s = WF^t + B$
Texture variation
STM
$\begin{pmatrix} x^s \\ y^s \end{pmatrix} = \begin{pmatrix} 1 & 0 & a \\ 0 & 1 & b \end{pmatrix} \begin{pmatrix} x^t \\ y^t \\ 1 \end{pmatrix}$
Translate $(a \quad b)$
STM
$\begin{pmatrix} x^s \\ y^s \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{pmatrix} \begin{pmatrix} x^t \\ y^t \\ 1 \end{pmatrix}$
Rotate θ
CNN
CNN
CNN
Positive Samples
Test Samples
Results

Fig. 1. Background-Adaptive Surface Defect Detection Neural Networks via Positive Samples. We introduce a foreground edge attention mechanism (FEAM) to mitigate texture variation noise and a spatial transformer module (STM) to counteract translation and rotation noise.

Despite the progress made in deep learning algorithms, surface defect detection continues to pose a significant challenge due to several factors. These include the high resemblance between defect and non-defect areas, the minuscule size of defects, intra-class inconsistency, and inter-class indistinction. The advent of sophisticated algorithms like PGANet [1] and AIS-Net [2] has successfully addressed some of these challenges, utilizing attention mechanisms and multi-scale feature fusion to achieve unparalleled precision. However, these methods presuppose identical distribution of training and

*Tongzhi Niu and Biao Chen contributed equally to this work. **Bin Li is corresponding authour.

testing datasets - a scenario rarely met in some real-world applications due to continuous and batch-specific changes in sample backgrounds.

The trend towards flexible production lines catering to small batches and product variety leads to substantial variations between different batches. These dynamic conditions underscore the need for innovative solutions that can accommodate such variability, ensuring reliable performance. As a response, methods based on Few-shot learning and domain adaptation have been introduced for defect detection tasks. These methods fall into two categories: foreground (defect) adaptation and background (defect-free) adaptation. Foreground adaptation, as exemplified by TGRNet's novel C-way N-shot W-normal method [3], constructs graph-structured data based on positive sample background features and few-shot sample foreground features, achieving generalization to new defect types. Background-adaptation, represented by Shuai et al.'s one-shot unsupervised domain adaptation framework (OUDA) [4], effectively facilitates surface defect detection across different Rails types.

This paper primarily focuses on background-adaptation. Given the inherent imbalance in defect detection data, positive samples for new batches are usually more readily available. Hence, existing methods [3], [5], [6] often emphasize learning the ability to compare testing samples with positive samples, promoting background adaptation based on these positive samples. This approach promises to enhance the generalizability and applicability of defect detection systems across varied real-world scenarios.

The challenge lies in distinguishing between true defects and variations caused by noise, as differences between testing and positive samples extend beyond defect features. We categorize this noise into texture variation noise, translation noise, and rotation noise. Existing methods have approached the noise issue in various ways. For instance, Sianmese Unet [5] subtracts the features of the positive sample directly from the test sample to obtain anomaly features, but this method does not take noise interference into account. On the other hand, OUDA [4] uses style transfer to remove texture variation noise, achieving detection of different types of rail defects. DSSSnet [6] proposes a kind of class-max pooling method to suppress noise, enabling the detection of PCB defects. However, these methods do not separately discuss different types of noise or delve into the forms of noise and methods for their elimination.

In response, we propose a novel framework, Background-Adaptive Surface Defect Detection Neural Networks via Positive samples (BANet), designed to eliminate different noise types. We introduce a foreground edge attention mechanism (FEAM) to mitigate texture variation noise and a spatial transformer module (STM) to counteract translation and rotation noise.

First, texture variation noise emerges as patterns of pixel value shifts within the backdrop. To address this, we propose a FEAM, inspired by DFNNet [7]. Our aim with FEAM is to enhance the network's proficiency in differentiating foreground from background areas. This mechanism, which operates across low to high levels, is capable of concurrently acquiring precise edge details from low-level features and procuring semantic data from high-level ones, allowing for better generalization across diverse background texture variations.

Second, translation and rotation noise manifest as alterations in the relative positioning of background features relative to positive samples, incorporating both shifts and rotations. Given the innate invariance of Convolutional Neural Networks (CNNs) to translation and rotation, these types of noise present formidable obstacles to elimination. Taking a cue from Spatial Transformer Networks (STNs) [8], we propose a STM to address these types of noise based on affine transformations. This innovative module is designed to counteract the perturbing impacts of these noise types, thereby bolstering the model's precision in real-world defect detection scenarios

In summary, this paper contributes:

- A Background-Adaptive Surface Defect Detection Network enhancing generalization capacity by learning the comparison ability between positive and testing samples.
- A Foreground Edge Attention Mechanism (FEAM) designed to enhance the model's ability to distinguish between the foreground and background and effectively eliminate texture variation noise.
- A Spatial Transformer Module (STM) based on affine transformations, eliminating translation and rotation noise.
- Empirical validation of our model on Optical Communication Devices (OCDs) dataset, demonstrating superior performance and potential for practical applications.

II. Related Works

The work presented in this paper draws upon two major areas in deep learning: attention mechanisms and domain adaptation. Here we review the most pertinent work in these fields, particularly as they relate to defect detection in industrial manufacturing contexts.

A. Attention Mechanisms

The concept of attention mechanisms, initially inspired by human visual attention, has been widely incorporated in deep learning models to help them focus on relevant features and ignore irrelevant ones.

Recurrent Models of Visual Attention (RAM) processes inputs sequentially, deciding where to look next based on past observations [9]. Spatial Transformer Networks (STN) actively spatially transform feature maps, providing invariance to translation, scale, rotation, and other affine transformations [8]. Squeeze-and-Excitation Networks boost the performance of CNNs by explicitly modeling the interdependencies between the channels of convolutional features [10]. Non-local Neural Networks capture long-range dependencies based on the self-attention mechanism, useful in video understanding and 3D tasks [11]. Vision Transformer applies Transformers directly to image patches, treating an image as a sequence of patches [12]. Swin Transformer applies transformers to non-overlapping windows of features at different scales, reducing complexity and enabling broader application [13].

In surface defect detection, PGA-Net [1] design a global context attention module, which embedded in these resolutions to ensure efficient information transfer from low-resolution to high-resolution. AIS-Net [2] present a dual attention context guidance module for achieving full utilization of global and local context information of defect feature maps, thereby capturing more information of tiny defects. RetinaNet with difference channel attention and adaptively spatial feature fusion is propsed for steel suface defect detection [14].

In our research, building upon the solid foundation of attention mechanisms as established in models such as DFNNet and STN, we propose and implement novel attention strategies — FEAM and STM.

B. Domain Adaptation

Domain adaptation addresses the problem of performance degradation that occurs when the distribution of training data differs from that of testing data. This is a major concern in many real-world applications, where the model needs to generalize well across different contexts.

The evolution of domain adaptation techniques started with the development of the Domain Adversarial Neural Network (DANN), which introduced a domain adversarial loss to overcome the challenge of domain shift, enhancing model generalization across different feature distributions [15]. This was followed by the introduction of Cycle-Consistent Generative Adversarial Networks (CycleGAN), which innovatively utilized cycle-consistent adversarial loss for unpaired image-to-image translation, allowing transformations between distinct domains without requiring paired training examples [16]. While this approach has the advantage of bypassing the need for labeling target domain images, it hinges on the availability of an ample volume of target domain samples. Though this method eliminates the need for target domain image labeling, it relies on the ample availability of such samples. Given the scarcity of defect samples, collecting adequate target domain data can be even more challenging than the labeling effort itself.

The ASM [17] framework introduced a solution to the paucity of target domain samples through a one-shot unsupervised domain adaptation approach. This was achieved by ingeniously integrating the style transfer and task-specific modules in an adversarial manner. Similarly, the OUDA [4] method proposed an innovative shape-consistent, one-shot, unsupervised domain adaptation strategy, designed specifically to mitigate performance degradation associated with domain shifts.

Building upon the existing methodologies, our work pivots around meticulously addressing the noise discrepancies between positive and test samples. This includes an in-depth exploration and categorization of various types of noise, followed by the proposition of novel methodologies for their effective alleviation.

III. METHODOLOGY

A. Problem Definition

In this paper, we focus on the problem of background-adaption. It is assumed to have access to the source domain $D_s = (x_i^s, \hat{x}_i^s, y_i^s)_i^N$, where x_i^s denote the samples, $\hat{x}_i^s$ denote the positive samples, and both x_i^s and corresponding $\hat{x}_i^s$ belong to the same batch. And only positive samples $D_t = (\hat{x}_i^t)_i^N$ is available for domain adaptation. The goal of background-adaption is to use these samples to train a model that accurately segments defect in the target domain. It is worth noting that the source and target domain foreground features belong to the same domain, while the background features have domain shifts.

B. Noise Types and Definition

This paper primarily addresses the challenge of noise mitigation during the comparative analysis between positive and test samples to effectively extract the defect features. To facilitate understanding, we represent features with a 3×3 matrix. The feature matrix for the positive sample $F_{\hat{x}_i}$ (without noise) and defect sample $F_{x_i}^d$ (without noise) are as follows:

$$F_{\hat{x}_i} = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & f_{2,2} & f_{2,3} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \tag{1}$$

herein, $f_{i,j}$ denotes the local features at coordinates (i, j), where i, j can each take on the values 0, 1, or 2.

$$F_{x_i}^d = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ f_{2,1} & \underline{\boldsymbol{d_{2,2}}} & \underline{\boldsymbol{d_{2,3}}} \\ f_{3,1} & f_{3,2} & f_{3,3} \end{bmatrix} \tag{2}$$

We propose the following categorization for noise:

1) Texture Variation Noise: This noise category is characterized by alterations in the product's surface attributes, predominantly due to the batch-to-batch variations in the product constituents. The feature matrix of test sample with texture variation noise (without defect) is:

$$F_{x_i}^v = \begin{bmatrix} \lambda(f_{1,1}) & \lambda(f_{1,2}) & \lambda(f_{1,3}) \\ \lambda(f_{2,1}) & \lambda(f_{2,2}) & \lambda(f_{2,3}) \\ \lambda(f_{3,1}) & \lambda(z_{3,2}) & \lambda(f_{3,3}) \end{bmatrix} \tag{3}$$

2) Translation Noise: This type of noise emerges from displacements of background elements or workpieces, leading to shifts from their initial positioning. The feature matrix of test sample with translation noise (without defect) is:

$$F_{x_i}^t = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} \\ \boldsymbol{f_{3,1}} & \boldsymbol{f_{3,2}} & \boldsymbol{f_{3,3}} \\ \boldsymbol{f_{2,1}} & \boldsymbol{f_{2,2}} & \boldsymbol{f_{2,3}} \end{bmatrix} \tag{4}$$

3) Rotation Noise: This category of noise is ascribed to the rotational movements of background elements, causing deviations from their default orientation. The feature matrix of test sample with rotation noise (without defect) is:

$$F^r_{x_i} = \begin{bmatrix} f_{1,1} & \boldsymbol{f_{2,3}} & f_{2,3} \\ \boldsymbol{f_{1,2}} & f_{2,2} & \boldsymbol{f_{3,2}} \\ f_{3,1} & \boldsymbol{f_{2,1}} & f_{3,3} \end{bmatrix} \tag{5}$$

C. Overview

In this paper, we propose BANet, which doesn't directly learn the representational capability of samples, but rather develops an adaptive detection for different product batches by learning the contrastive ability between positive and test samples. The network architecture resembles a Siamese network, as shown in Fig. 2. It employs a Foreground Edge Attention Mechanism (FEAM) to alleviate texture variation noise and a Spatial Transformer Module (STM) to address translation and rotation noise. To further enhance network training, we introduce a deep supervision loss function in the feature decoding segment. Subsequently, we will provide detailed descriptions of the FEAM, STM, and the loss function.

D. Foreground Edge Attention Mechanism (FEAM)

In the case of texture variation noise, the inherent robust feature representation potential of neural networks could directly mitigate such noise. Therefore, we designed a FEAM to enhance the network's ability to distinguish between the foreground and background. FEAM operates by directly learning a semantic boundary under explicit semantic boundary supervision, mirroring the characteristics of a semantic boundary detection task. This approach facilitates distinguishing features on either side of the semantic boundary, enhancing the network's sensitivity to nuances between foreground and background.

As shown in Fig. 3, the FEAM, functioning in a stage-wise manner, is capable of concurrently extracting accurate edge information from low-level features and semantic information from high-level features. This approach helps to compensate for the lack of semantic information in the original edges. The incorporation of high-level semantic information refines the detailed edge information extracted from the lower stages. The network's supervisory signal is derived from the ground truth of the semantic segmentation through the application of traditional image processing techniques, such as the Canny method.

In this context, the Loss function $L_{assist1}$ we utilize is the Binary Cross-Entropy Loss (BCELoss), which is specifically formulated as follows:

$$\begin{aligned} L_{assist1}(p, y) = &- \frac{1}{N} \sum_{i=1}^{N} Canny(y_i) \log(p_i) \\ &+ (1 - Canny(y_i)) \log(1 - p_i) \end{aligned} \tag{6}$$

where N is the total number of samples. y_i is the true label for sample i. p_i is the predicted probability of observation i obtained by FEAM. $Canny(y_i)$ is the Canny operator applied to the true label of sample i.

In addition, we enhanced our network's capabilities by employing deep supervision within the decoder. The associated loss $L_{assist2}$ function is defined as follows:

$$\begin{aligned} L_{assist2}(p, y) = &- \frac{1}{N} \sum_{i=1}^{N} y_i \log(p_i) \\ &+ (1 - y_i) \log(1 - p_i) \end{aligned} \tag{7}$$

where p_i represents the concatenated features from all layers of the decoder, which are subsequently processed through a 3×3 convolution operation.

E. Spatial Transformer Module (STM)

In the face of displacement and rotation noise, the position of the features changes. Consequently, we aim to correct the coordinates of the feature positions through affine transformations. Inspired by the STN [8], our proposed STM follows the structure illustrated in Figure 4, which primarily comprises the Localization Network and Grid Generator components.

1) Localization network: The localization network processes the input feature map $F_{x_i}, F\hat{x}_i \in \mathbb{R}^{H \times W \times C}$, which exhibits a width W, height H, and channels C. The network subsequently generates θ, the parameters that prescribe the transformation τ_θ to be enacted on the feature map: $\theta = f_{loc}(F_{\hat{x}_i})$. As illustrated in equation 8, the dimensions of the affine transformation θ amount to six.

The function $f_{loc}()$ of the localization network constitutes a convolutional network, further incorporating a fully-connected network in its final regression layer, designed to generate the transformation parameters θ.

2) Grid generator: In our study, we apply affine transformations to multiple feature layers. We define the output features to rest on a regular grid $G = \{G_i\}$, where each feature G_i corresponds to (x^t_i, y^t_i). For the sake of clarity, assuming that τ_θ represents a 2D affine transformation A_θ, we can express the pointwise transformation as follows:

$$\begin{pmatrix} x^s_i \\ y^s_i \end{pmatrix} = \tau_\theta(G_i) = A_\theta \begin{pmatrix} x^t_i \\ y^t_i \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x^t_i \\ y^t_i \\ 1 \end{pmatrix} \tag{8}$$

where (x^t_i, y^t_i) are the target coordinates of the regular grid in the output feature map, (x^s_i, y^s_i) are the source coordinates in the input feature map that define the sample points.

We employ normalized coordinates in terms of height H and width W, ensuring the transformed features fall within the spatial bounds of the output. The transformation and sampling process aligns with the standard texture mapping and coordinate usage in graphic processing.

Defined by equation 8, the transformation enables translation, and rotation of the input feature map. This requires merely six parameters (the six elements of A_θ) to be produced by the localization network.

If we aim to apply a translation of a units along the x-direction and b units along the y-direction, the parameter transformation can be represented as follows:

$$A_\theta = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \end{bmatrix}$$

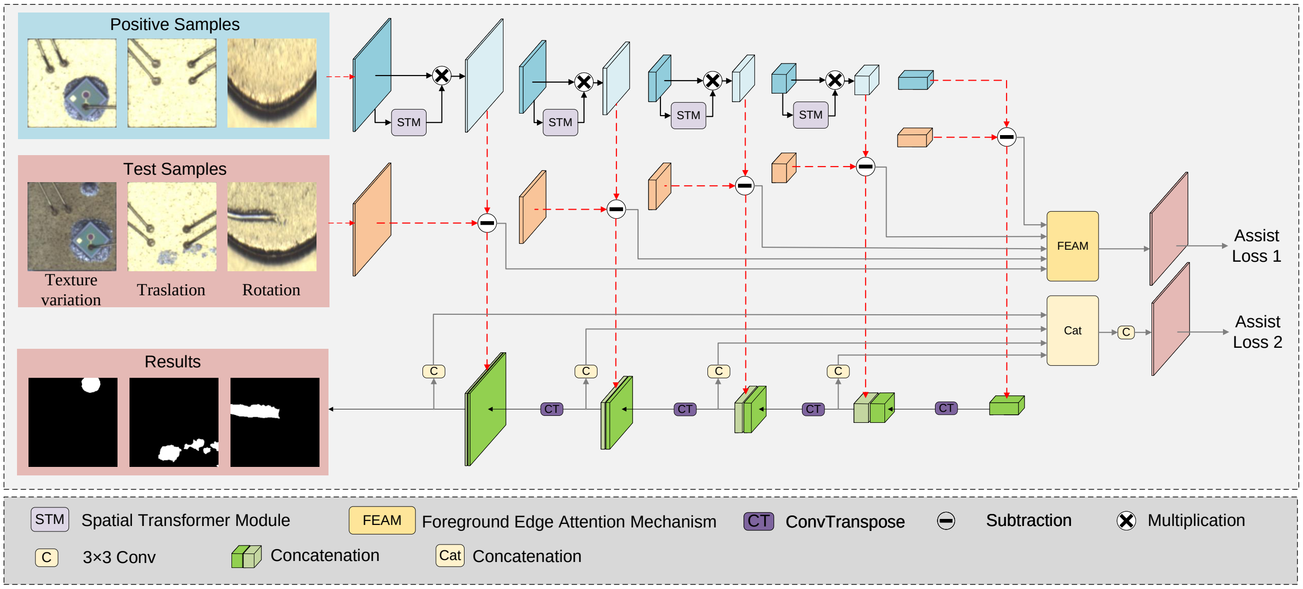

Fig. 2. The overview of proposed Background-Adaptive Surface Defect Detection Network (BANet)

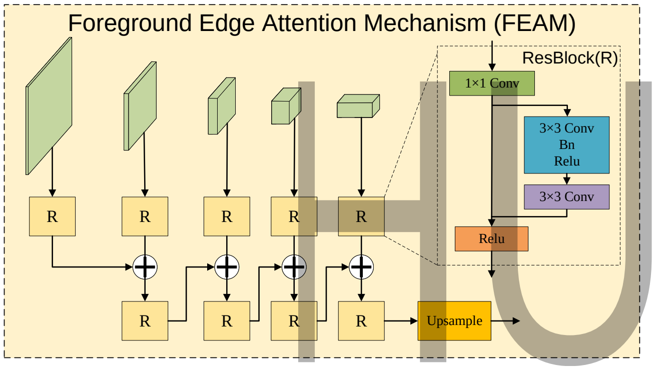

Fig. 3. Foreground Edge Attention Mechanism (FEAM)

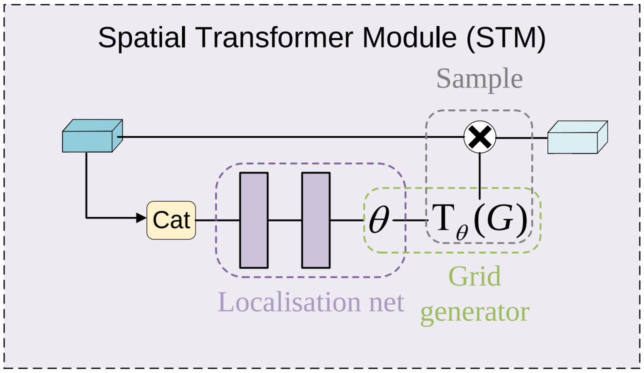

Fig. 4. Spatial Transformer Module (STM)

In this case, the transformation matrix A_θ would shift each point in the feature space by a units in the x-direction and b units in the y-direction, effectively translating the entire feature map.

If the objective is to rotate the image by an angle of θ, the transformation parameters are represented as:

$$A_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{bmatrix}$$

This rotation matrix will rotate each point in the image counter-clockwise by an angle θ around the origin of the coordinate system.

F. Loss Function

In the revised version:

Managing pixel-level defects can be challenging due to their typically sparse occurrence and the resultant imbalanced distribution between positive and negative class pixels. To tackle this imbalance, we incorporated the focal loss function [18] into our model. This function places a greater emphasis on hard-to-segment and mis-segmented pixels, helping to alleviate the issues stemming from data imbalance. The focal loss is defined as follows:

$$L_{local} = -\alpha(1-p_t)^\gamma \log(p_t) \tag{9}$$

In the equation above, $p_t \in [0, 1]$ represents the probability obtained by GWNet. $\gamma \geq 0$ is a modifiable focusing parameter; when set to 0, the focal loss is equivalent to cross entropy loss. As the value of γ increases, the effect of the modulating factor is similarly amplified (we utilized $\gamma = 2$ in our model). $\alpha \in [0, 1]$ serves as a weighting factor to counteract class imbalance (in our model, we set $\alpha = 0.25$).

Therefore, according to equation 6, 7, and 9, the final loss L_{final}:

$$L_{final} = (1-2\alpha)L_{local} + \alpha L_{assist1} + \alpha L_{assist2} \tag{10}$$

TABLE I
QUANTITATIVE COMPARISON WITH STATE-OF-ARTS METHODS

	Method	***Pre***	***Recall***	***F1***	***mIoU***	**Params(MB)**	**Flops(G)**
Classical segmentation methods	U-Net	0.8597	0.6926	0.7590	0.6325	7.68	14.27
	FCN	0.8831	0.7376	0.8038	0.6580	22.35	10.28
	SegNet	0.8949	0.3907	0.5440	0.3662	40.47	29.45
	DeepLabV3+	0.8295	0.7967	0.8128	0.6702	59.47	24.09
	PGANet	**0.9186**	0.4793	0.6299	0.4483	51.40	51.50
Attention-based methods	CCNet	0.8224	0.3875	0.5268	0.3614	67.70	39.18
	DUNet	0.8716	0.3100	0.4574	0.2942	13.58	35.11
	DANet	0.8220	0.5748	0.6765	0.5130	47.46	14.76
	Swin-U-Net	0.6612	0.2569	0.3700	0.2076	27.15	7.74
Unsupervised domain adaption methods	Siamese U-Net	0.8913	0.6946	0.7807	0.6243	7.85	18.53
	DSSSNet	0.8931	0.8148	0.8521	0.7405	6.10	14.36
Ours	BANet	0.8827	**0.9214**	**0.8935**	**0.8146**	44.08	45.53

In this equation, α is a hyperparameter which is set to 0.25 in our work.

IV. EXPERIMENTS

A. Implementation details

BANet is instantiated on the PyTorch framework, using a single NVIDIA Tesla V100 for computation. For the training process, we deploy the Adam optimizer, maintaining a batch size of 16 and utilizing a learning rate of 0.00001. This configuration offers an optimal balance between resource usage and network performance.

B. Evaluation Metrics

To conduct a thorough comparison of various methods, we apply five widely recognized metrics used for semantic segmentation performance evaluation: Precision (Pre), Recall (Rec), F-measure (F1), and mean Intersection over Union (mIoU).

The mIoU is a particularly pertinent evaluation metric for semantic segmentation, gauging the extent of overlap between the predicted and ground truth labels. The F-measure, a harmonic mean of precision and recall, provides a comprehensive reflection of the performance in binary semantic segmentation tasks. The formal definitions of these metrics are as follows:

$$Pre = \frac{TP}{TP + FP} \tag{11}$$

$$Rec = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = \frac{2 \times Pre \times Rec}{Pre + Rec} \tag{13}$$

$$mIoU = \frac{TP}{FP + FN + TP} \tag{14}$$

where, TP, FP, TN, and FN signify the count of true positives, false positives, true negatives, and false negatives respectively.

C. Comparison with the state-of-the-art model

We compared our method GWNet with eleven state-of-the-art methods, including five classical segmentation methods (U-Net [19], FCN [20], SegNet [21], DeepLabV3+ [22]), and PGANet [1], four attention based segmentation methods (CCNet [23], DUNet [24], DANet [25]) and Swin-U-Net [26], and two unsupervised domain adative methods (DSSSNet [6], Siamese U-Net [5]). And these methods are compared on OCDs dataset. OCDs dataset The OCDs dataset is collected from the flexible production line of optical communication devices, with obvious characteristics of small batches and multiple types. And there are noises between inputs and templates, including texture variation, translation, and rotation noise.

1) Quantitatively Analysis: As depicted in Table I, we observe that among the traditional segmentation methods, DeepLabV3+ exhibits superior performance, demonstrating remarkable generalization capabilities. Attention-based methods overall exhibit subpar performance, indicating that standalone attention mechanisms provide minimal assistance for domain adaptation. With the incorporation of comparison-oriented strategies, while the Siamese U-net, which does not handle noise, shows no noticeable improvement over U-net, DSSSNet makes a substantial leap in performance after processing noise through class-max pooling. Compared to DSSSNet, our method boosts the mIoU by 7.41%, confirming the effectiveness of our approach in handling noise. At last, while our proposed approach indeed involves a higher level of computational complexity and parameter quantity, it correspondingly delivers exceptionally robust results.

2) Qualitatively Analysis: As illustrated in Figure 5, we notice that traditional and attention-based methods are highly susceptible to noise. In contrast, DSSSNet outperforms other methods to a degree. However, its noise handling capabilities still fall short compared to BANet, particularly regarding displacement and rotation noise. Pooling operations combined with convolution operations ensure a certain degree of translational invariance for the network, but this implicit invariance is limited. By explicitly implementing affine transformations, we have successfully eliminated displacement and rotation noise, yielding exceptionally satisfying results.

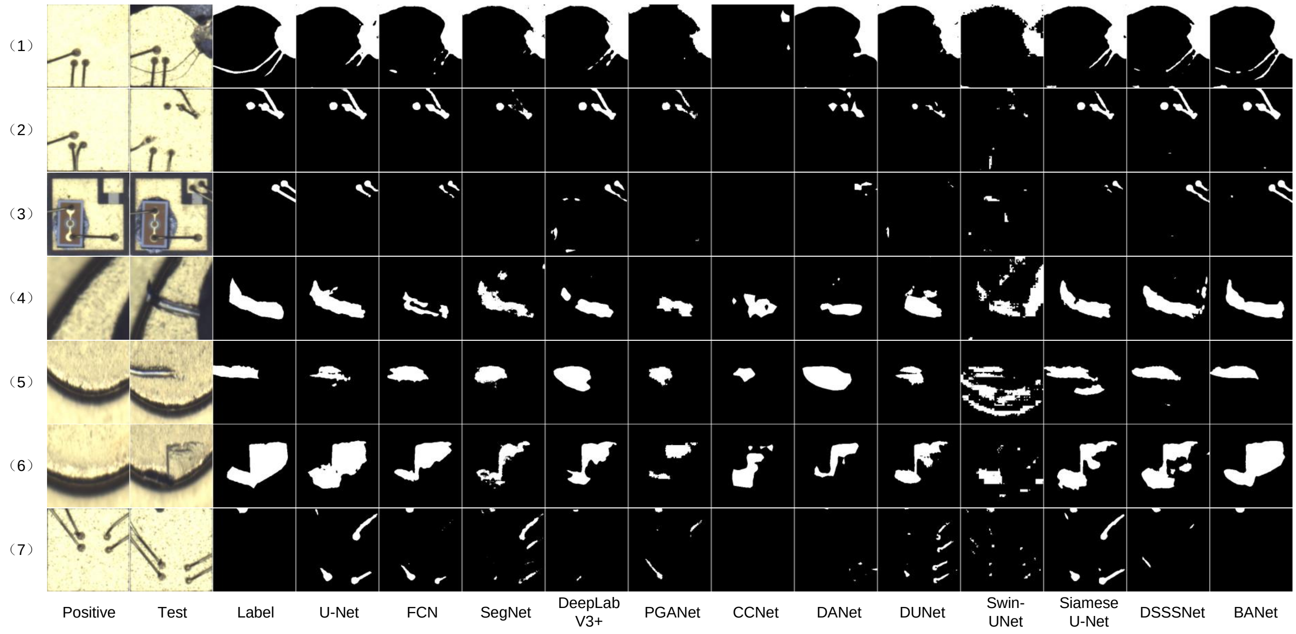

Fig. 5. Visual comparison with state-of-the-arts methods

D. *Ablation studies and discussion*

In the ablation studies, we concatenate the positive samples and test samples along the channel direction and directly input them into the U-Net network with skip connections as the baseline. This is compared to a Siamese network with FEAM and STM components removed. This process produces the results shown in Table II and Figure 6.

TABLE II
RESULTS OF ABLATION

Modules	Baseline	Siamese	EFAM	STM	mIoU	F1
S1	✓				0.6766	0.8102
S2	✓	✓			0.6975	0.8246
S3	✓	✓	✓		0.8022	0.8858
S4	✓	✓		✓	0.8051	0.8871
S5	✓	✓	✓	✓	0.8146	0.8935

As illustrated in Table II, in comparison to the two fundamental networks, our approach significantly improved detection performance, with an mIoU increase of 11.71% and a F1 Score increase of 6.89%. Figure 6, which uses class activation to display the feature map, shows that our method successfully eliminates texture variation, displacement, and rotation noise, accurately extracting defect features.

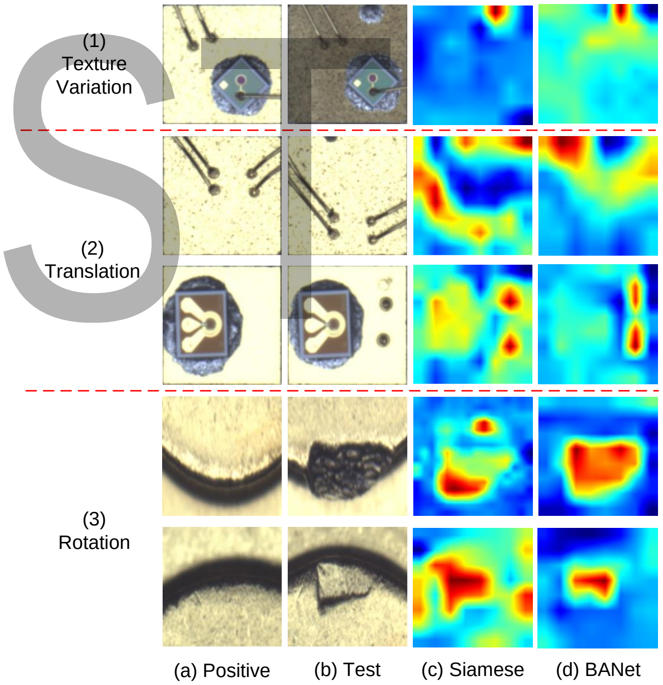

Fig. 6. Visual Result of Ablation

V. Conclusion

In this paper, we have addressed the crucial issue of defect detection in an ever-evolving industrial landscape. We developed the Background-Adaptive Surface Defect Detection Neural Networks via Positive samples (BANet), which tackles the problem of texture variation, translation, and rotation noise. Through our Foreground Edge Attention Mechanism (FEAM) and Spatial Transformer Module (STM), our proposed model demonstrates an improved ability to distinguish between foreground and background and manage different types of noise.

The empirical testing of our model on the Optical Communication Devices (OCDs) dataset validated its superior performance and potential practical applicability. This paper has not only made significant strides in surface defect detection but also sets a promising course for future research in this dynamic field. Further work will focus on enhancing the adaptability and generalization capacity of our model to meet the challenges of the constantly changing manufacturing environment.

References

[1] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7448–7458, 2019.

[2] T. Liu, Z. He, Z. Lin, G.-Z. Cao, W. Su, and S. Xie, "An adaptive image segmentation network for surface defect detection," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[3] 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.

[4] S. Ma, K. Song, M. Niu, H. Tian, Y. Wang, and Y. Yan, "Shape consistent one-shot unsupervised domain adaptation for rail surface defect segmentation," *IEEE Transactions on Industrial Informatics*, 2023.

[5] D. Kwon, J. Ahn, J. Kim, I. Choi, S. Jeong, Y.-S. Lee, J. Park, and M. Lee, "Siamese u-net with healthy template for accurate segmentation of intracranial hemorrhage," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*. Springer, 2019, pp. 848–855.

[6] Z. Ling, A. Zhang, D. Ma, Y. Shi, and H. Wen, "Deep siamese semantic segmentation network for pcb welding defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[7] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1857–1866.

[8] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[9] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," *Advances in neural information processing systems*, vol. 27, 2014.

[10] 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.

[11] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[14] X. Cheng and J. Yu, "Retinanet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2020.

[15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[16] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[17] Y. Luo, P. Liu, T. Guan, J. Yu, and Y. Yang, "Adversarial style mining for one-shot unsupervised domain adaptation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 612–20 623, 2020.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[19] 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.

[20] 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.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[22] 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.

[23] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. Huang, "Ccnet: Criss-cross attention for semantic segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[24] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, "Dunet: A deformable network for retinal vessel segmentation," *Knowledge-Based Systems*, vol. 178, pp. 149–162, 2019.

[25] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.

[26] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*. Springer, 2023, pp. 205–218.