Feature Matching Driven Background Generalization Neural Networks for Surface Defect Segmentation

Biao Chen[a], Tongzhi Niu[a,*], Ruoqi Zhang[b], Hang Zhang[a], Yuchen Lin[a], Bin Li[a,c]

[a] *School of Mechanical Science and Engineering, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuahn, 430074, Hubei, China*
[b] *China-EU Institute for Clean and Renewable Energy, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuahn, 430074, Hubei, China*
[c] *Wuhan Intelligent Equipment Industrial Institute Co., Ltd, 8 Ligou South Road, Wuahn, 430074, Hubei, China*

Abstract

In this paper, we tackle the challenge of background generalization in surface defect segmentation for chips of surface-mounted devices, specifically in template-sample comparison algorithms where background features in templates and samples exhibit spatial variations such as shifts and rotations. A novel Background Generalization Network (BGNet) that utilizes a feature matching algorithm is introduced. BGNet begins with obtaining dense features filled with global and interactive information through a Siamese network and employing self- and cross-attention mechanisms from Transformers. Subsequently, the matching score is derived from feature similarity, and matching relations are determined via the Mutual Nearest Neighbor algorithm. Using these relations, we mitigate noise caused by spatial variations and implement a multi-scale fusion of detail and semantic information, which leads to accurate segmentation results. Our experiments on OCDs and PCBs datasets demonstrated that BGNet outperforms state-of-the-art methods.

Keywords:
Surface Defect Detection, Neural Networks, Feature Matching, Background Generalization

*corresponding author

1. Introduction

In recent years, surface defect detection technology based on deep learning has been widely researched and applied in industries such as semiconductors and electronics [1], automotive [2], transportation [3], and textiles [4]. Extensive and comprehensive studies have addressed challenges such as data imbalance [5], multiple scales and shapes [6, 7], and significant intra-class variations versus minor inter-class differences [8]. This paper focuses on an emerging issue: achieving batch-to-batch background generalization through template-sample comparison [9, 10, 11].

The success of traditional deep learning relies on the assumption that training and testing datasets share the same distribution. However, in the field of surface defect detection for chips of surface-mounted devices (including Printed Circuit Boards (PCBs) and Optical Communication Devices (OCDs), etc), variations in device types and distribution across different batches can lead to distinct data distributions. If defects are defined as the foreground and non-defects as the background, then the challenge of distribution inconsistency can be described as background generalization. Existing methods [9, 10, 11] aim to learn how to compare the changes between the template and the samples. For new batches, generalization can be achieved by collecting templates. The primary challenge with this approach is the noise that arises from inconsistencies in device and fabrication processes, which contribute to the background variations between template and samples, beyond just the defect foreground features.

Typically, due to variations in device types and fabrication processes, the background features of templates and samples may exhibit spatial variations such as rotation, displacement, and dilation, as well as texture variations like color changes. For texture variations, existing convolutional neural networks have a powerful ability to extract features and demonstrate robust performance. However, when it comes to spatial variations, the spatial invariance of methods based on convolutional neural networks is limited [12].

To solve the challenge of spatial variations, existing methods employ activation functions, pooling operations, and attention mechanisms. The Siamese U-Net[13] directly subtracts the template from the sample at different feature layers and calculates the difference into an attention map using activation functions. DSSSNet [9] achieves limited spatial invariance through global pooling, which is pre-defined for dealing with variations in the spatial arrangement of data. GWNet [11] incorporates self-attention and

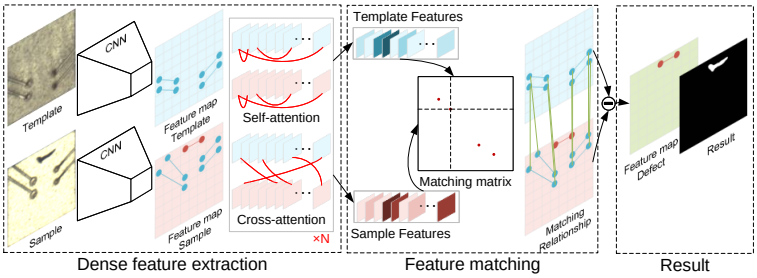

Figure 1: Feature Matching Driven Background Generalization Neural Networks for Surface Defect Segmentation

inter-attention mechanisms from Transformers [14] into CNNs. Based on the position-independent characteristics of features, it achieves spatial invariance and enables background generalization. Although these methods have shown some effectiveness, their operational mechanisms and design rationale are not always clear.

In this paper, a transparent and accountable Background Generalization Network (BGNet) based on feature matching is introduced, as illustrated in Figure 1. Despite spatial variations in displacement, rotation, and mapping among the background features of the template and sample, a one-to-one correlation persists. Therefore, a naive idea is to compute the matching relationship and adjust feature positions accordingly, allowing for one-to-one subtraction and achieving background generalization.

Given the unavailability of interest point labels, obtaining dense features equipped with matching information through a Convolutional Neural Networks-based Siamese network is an initial step. However, these dense features have a limited receptive field and lack interrelated information between the template and sample. To address this, existing detect-free local feature matching methods [15, 16, 17, 18] are employed, using a Transformer [14] to obtain global context through self-attention and interrelated information through cross-attention. Multiple iterations of self- and cross-attention are layered for more robust representation.

After obtaining dense features of both the template and sample, similarity measures, as recommended in [19], are employed to calculate matching scores among feature points and derive a matching matrix. This matrix is preprocessed using a dual-softmax function [15] with a threshold for isolating and matching significant features. Matching relationships are ultimately determined using a mutual nearest neighbor strategy.

Based on the matching relationships derived, matching features are subtracted in the sample from those in the template to acquire noise-free defect foreground features. To achieve more accurate segmentation, a multi-scale fusion technique is employed. However, considering computational demands, the feature matching network is applied separately to detailed (1/32 scale) and semantic information (1/8 scale) within BGNet, as suggested by Bisenet [20].

BGNet's performance is evaluated using surface-mounted device chip datasets for both OCDs [11] and PCBs [9] datasets. Experimental results show that BGNet outperforms current state-of-the-art techniques. Visualization experiments demonstrate BGNet's ability to accurately identify features undergoing spatial transformations within the background. After successful matching and subtraction, only foreground defect features remain.

2. Related Work

2.1. Surface defect detection

In recent years, the issue of data imbalance in surface defect detection has received widespread attention. Anomaly detection [21] algorithms based on positive samples, data generation algorithms [22], and generalizable algorithms for new foreground (defect) types and background (defect-free) types have been extensively researched. This paper focuses on the study of generalizable algorithms and provides a detailed introduction to both foreground and background generalization.

2.1.1. Foreground generalization

In the field of surface defect classification, several novel approaches have been proposed. The Graph Embedding and Distribution Transformation (GEDT) model[23], in combination with the Optimal Transport (OPT) module, can identify new defect classes even with a limited number of labeled samples. The FSDR approach [24] advances a coarse-to-fine few-shot defect classification strategy that employs dynamic weighting and joint metrics, easing the data collection process and enabling classification of novel defect categories. FaNet [25] introduces a feature-attention convolution module that excels at extracting comprehensive feature details from base classes while enhancing semantic integration by capitalizing on long-range feature interconnections.

In the context of surface defect segmentation, several notable methodologies have emerged. TGRNet [26] applies few-shot learning theory to generic metal surface defect segmentation and devises a C-way N-shot W-normal learning method that includes a surface defect triplet to independently segment the background and defect areas. It also incorporates a multi-graph reasoning module to explore similarity relationships among different images. Simultaneously, OBFTNet [27] introduces background images as supplementary learning information and treats few-shot segmentation as an optimal bilateral transport problem, adaptively generating task-specific semantic correspondences to ensure the model's ability to generalize to unseen materials. Recently, a comparative dataset known as Industrial-5^i [28] has been constructed using public datasets.

2.1.2. Background generalization

In some flexible production lines, particularly with chips of surface-mounted devices, the types of defect foregrounds rarely increase, while the backgrounds vary with batch changes. As a result, background generalization is a valuable research topic.

DSSSNet [9] establishes a deep Siamese semantic segmentation network by combining the similarity measurement capabilities of the Siamese network with an encoder-decoder semantic segmentation network, resulting in an effective tool for PCB welding defect detection. Concurrently, SC-OSDA [10] presents a shape consistent style transfer module to address the issue of insufficient target domain samples by performing pixel-level distribution alignment between training and test images. This approach, requiring only a single target domain sample, significantly enhances the model's robustness to domain shifts. GWNet [11], introduces a Dual-Attention Mechanism (DAM) for the feature extraction and a Recurrent Residual Attention Mechanism (RRAM) for the feature fusion, enabling the model to effectively generalize to new batches of unseen data during training by utilizing collected templates.

In summery, adapting models to new defects or data is a significant challenge, with current methods still being explored and not yet ready for practical implementation. Given the consistent nature of defect features, background generalization is a more feasible and practical approach at this stage, particularly in the context of flexible production lines. This paper proposes an explicit and explainable method for this task, building upon prior research.

2.2. Local Feature matching

In general, local feature matching between images is the foundation of many 3D computer vision tasks, including structure from motion, simultaneous localization and mapping, and visual localization. Image matching methods typically use a three-stage process: feature detection, description, and matching. In the detection stage, significant points are identified in each image. Local descriptors are then extracted from the areas around these points. The result is two sets of descriptors, whose correspondences are established using nearest neighbor searches or advanced matching algorithms. Based on these stages, existing techniques can be divided into two categories: detector-based and detector-free local feature matching methods.

2.2.1. Detector-based local feature matching

Before the advent of deep learning, hand-crafted methods were often based on SIFT [29] and ORB [30]. SIFT characterizes distinctive keypoints by constructing a high-dimensional vector that represents the image gradients within a localized region of the image. ORB proposes an extremely fast binary descriptor based on BRIEF [31], offering speed that is two orders of magnitude faster than SIFT. Notably, both ORB and SIFT demonstrate rotation invariance and robustness to noise

Due to their powerful feature extraction capabilities, deep learning-based methods significantly improve performance under substantial viewpoint and illumination changes. LIFT [32] is the first to introduce an end-to-end differentiable complete feature point handling pipeline, including detection, orientation estimation, and feature description. Most recent research [33, 34, 35, 36] on deep learning for matching typically focuses on learning superior sparse detectors and local descriptors from data using Convolutional Neural Networks (CNNs).

However, methods based on CNNs typically use the nearest neighbor search to find matches among the extracted points of interest. SuperGlue [19] learns matches with a Graph Neural Network (GNN), which is a generalized form of Transformers [14]. Although SuperGlue demonstrates impressive performance, it fails to detect repeatable points of interest in indistinct regions.

2.2.2. Detector-free local feature matching

Detector-free methods bypass the feature detection phase and directly generate dense descriptors or dense feature matches. SIFT Flow [37] was the

first to propose pixel-wise SIFT features between two images while preserving spatial discontinuities.

In NCNet [38], exhaustive pairwise cosine similarities between two dense feature descriptors are computed and stored in a 4-D tensor, known as a correlation map. This map is then input into a neighbourhood consensus CNN (4D-CNN), which learns dense correspondences by regularizing the cost volume and enforcing neighborhood consensus among all matches. Following this line of work, Sparse-NCNet [39] employs sparse convolutions to improve efficiency. Moreover, DRC-Net [40] combines multi-scale information in a coarse-to-fine approach.

Similar to detector-based methods, the aforementioned detector-free methods also rely solely on local features to obtain descriptors. LoFTR [15], by utilizing both self- and cross-attention layers within the Transformer and repeatedly interleaving these layers, generates feature descriptors that are conditioned on both images, thereby learning densely arranged globally-consented matching priors inherent in the ground-truth matches. Transfusion [16] and Gmflow [17] also designed matching algorithms based on Transformer. However, these works rarely focus on the scale difference between the image pair. PATS [18] proposes patch area transportation with subdivision to obtain a significantly larger and more accurate number of matches.

This paper focuses on matching background features between templates and samples, which exhibit spatial variations. Despite the lack of interest points annotations, we have built upon previous research in detector-free local feature matching and proposed a background feature matching algorithm.

3. Methodology

3.1. Problem definition

This paper focuses on the challenge of background generalization, particularly in template-sample matching algorithms that deal with spatial variations in template and sample background features, including aspects such as translation, rotation and mapping. Given an image pair consisting of a template I^T and a sample I^S, they are input into a Siamese network, resulting in corresponding features at five different scales, denoted as $\left\{F_i^T\right\}_{i=1}^5$ and $\left\{F_i^S\right\}_{i=1}^5$. The feature map is represented as $F_i = (f_{x,y}) \in \mathbb{F}^{C\times H\times W}$, where C, H, W represent the channel, height, and width of the feature map F. The feature map of the template F_i^T, sample F_i^S, sample with translation $\hat{F}_i^S$, sample with rotation $\widetilde{F}_i^S$ are represented as follows:

$$F_i^T = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,W} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,W} \\ f_{3,1} & f_{3,2} & f_{3,3} & \cdots & f_{3,W} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{H,1} & f_{H,2} & f_{H,2} & \cdots & f_{H,W} \end{bmatrix} \tag{1}$$

$$F_i^S = \begin{bmatrix} f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,W} \\ f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,W} \\ f_{3,1} & \underline{\mathbf{d_{3,2}}} & \underline{\mathbf{d_{3,3}}} & \cdots & f_{3,W} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{H,1} & f_{H,2} & f_{H,2} & \cdots & f_{H,W} \end{bmatrix} \tag{2}$$

$$\hat{F}_i^S = \begin{bmatrix} f_{1,1} & \underline{\mathbf{f_{2,2}}} & f_{1,3} & \cdots & f_{1,W} \\ \underline{\mathbf{f_{3,1}}} & \underline{f_{1,2}} & f_{2,3} & \cdots & f_{2,W} \\ \underline{f_{2,1}} & d_{3,2} & d_{3,3} & \cdots & f_{3,W} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{H,1} & f_{H,2} & f_{H,2} & \cdots & f_{H,W} \end{bmatrix} \tag{3}$$

$$\widetilde{F}_i^S = \begin{bmatrix} \underline{\mathbf{f_{1,3}}} & f_{1,2} & \underline{f_{1,1}} & \cdots & f_{1,W} \\ \underline{\mathbf{f_{2,2}}} & \underline{f_{2,1}} & f_{2,3} & \cdots & f_{2,W} \\ \underline{\mathbf{f_{3,1}}} & d_{3,2} & d_{3,3} & \cdots & f_{3,W} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_{H,1} & f_{H,2} & f_{H,2} & \cdots & f_{H,W} \end{bmatrix} \tag{4}$$

where d represents the defective feature. The utilization of an underline (as seen in $\underline{f}$ and $\underline{d}$) indicates a change in feature location or type. The application of boldface (in $\mathbf{f}$ and $\mathbf{d}$) denotes the results of defects, translation, and rotation processes.

This study acknowledges that spatial variation, resulting in shifts in the positions of background features in both template and sample, renders direct subtraction ineffective. The primary focus of this paper is the development of a technique that matches these dynamic background features, enables corresponding subtractions, and thus produces more accurate segmentation results.

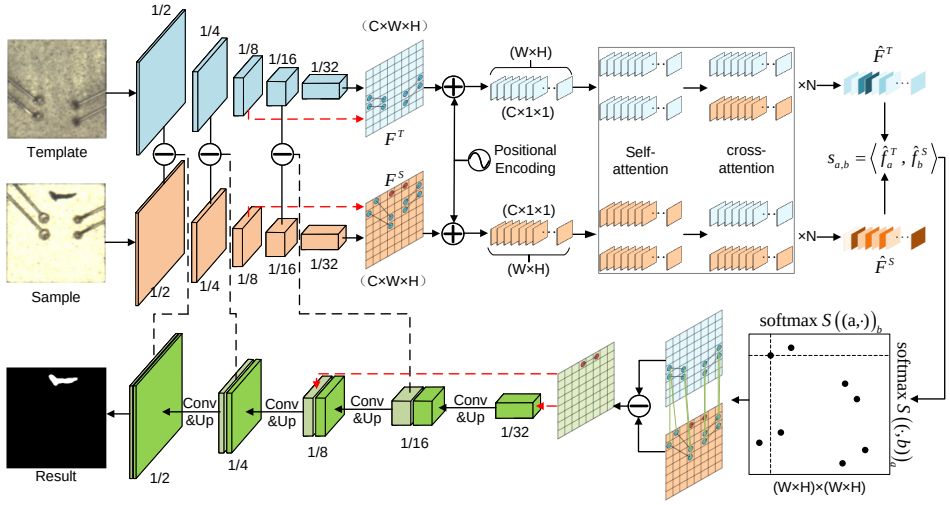

Figure 2: The BGNet architecture.

3.2. Framework overview

As show in Figure 2, BGNet consists of three components: Dense feature extraction, feature matching, and multi-scale feature fusion.

In the dense feature extraction section, the template image I^T and sample image I^S are input into the Siamese network to obtain feature maps of different scales $\left\{F_i^T\right\}_{i=1}^5$ and $\left\{F_i^S\right\}_{i=1}^5$, where $F_i = (f_{x,y}) \in \mathbb{F}^{C_i \times H_i \times W_i}$. Then, these feature maps are fed into the self-attention $Atten_{self}\left(\cdot,\cdot\right)$ and cross-attention $Atten_{cross}\left(\cdot,\cdot\right)$ mechanisms to yield dense feature maps, denoted as $\hat{F}_i^T$ and $\hat{F}_i^S$.

$$\hat{F}_i^T = Atten_{cross}\left(Atten_{self}\left(F_i^T, F_i^T\right), F_i^S\right) \tag{5}$$

$$\hat{F}_i^S = Atten_{cross}\left(Atten_{self}\left(F_i^S, F_i^S\right), F_i^T\right) \tag{6}$$

In the feature matching section, a proprietary feature matching algorithm $Match\left(\cdot,\cdot\right)$ is implemented to discern the matching relationship between the template and the sample, represented as $\mathcal{F}_i^M : M_i^S \rightarrow M_i^T$, where $M_i^S = \left\{\left(x^S, y^S\right)_j\right\}_{j=1}^N$ and $M_i^T = \left\{\left(x^T, y^T\right)_j\right\}_{j=1}^N$. Here, N denotes the number of matching features.

$$\mathcal{F}_i^M = Match\left(\hat{F}_i^T, \hat{F}_i^S\right) \tag{7}$$

Then, the noise-free features $F_i^D = \left(f_{x,y}^D\right) \in \mathbb{F}^{C_i \times H_i \times W_i}$ is eliminated by utilizing the matching relationship for corresponding subtractions.

$$f_{x,y}^D = \begin{cases} f_{x^S,y^S}^S & x^S, y^S \notin M_i^S \\ f_{x^S,y^S}^S - f_{x^T,y^T}^T & x^S, y^S \in M_i^S \end{cases} \tag{8}$$

In the multi-scale feature fusion, In the multi-scale feature fusion section, considering computational complexity, the feature matching network is applied independently to detailed (1/32 scale) and semantic information (1/8 scale). Concurrently, a direct subtraction is performed for other scales. Lastly, multi-scale feature fusion is achieved via a skip-connection approach.

3.3. Dense feature extraction

3.3.1. Siamese Network

This study employs a Siamese network to extract features from both the template and sample, consisting of two subnetworks with shared weights. Resnet-18 is used as the subnetwork, and pre-training weights based on ImageNet are utilized during the training process. The template I^T and sample I^S are input into the Siamese network, resulting in corresponding features at five different scales $\left\{F_i^T\right\}_{i=1}^5$ and $\left\{F_i^S\right\}_{i=1}^5$.

3.3.2. Positional encoding

In contrast to convolutional neural networks, Transformers input the entire feature map simultaneously, leading to the loss of inherent positional information in the image. To ensure appropriate matching of background features, this study augments the feature map with positional encoding information. Unlike previous methodologies, this work is focused on adding positional encoding to two-dimensional feature maps. Consequently, the positional encoding is defined as follows:

$$p_{x,y}^{(c)} = \begin{cases} sin\left(x \times \omega_k\right) & c = 4k \\ cos\left(x \times \omega_k\right) & c = 4k+1 \\ sin\left(y \times \omega_k\right) & c = 4k+2 \\ cos\left(y \times \omega_k\right) & c = 4k+3 \end{cases} \tag{9}$$

where $\omega_k = \frac{1}{10000^{2k/C}}$, $k = 0, 1, \cdots$. $p_{x,y} \in \mathbb{F}^C$ signifies the (x, y) positional encoding, while c stands for the dimension of the channel.

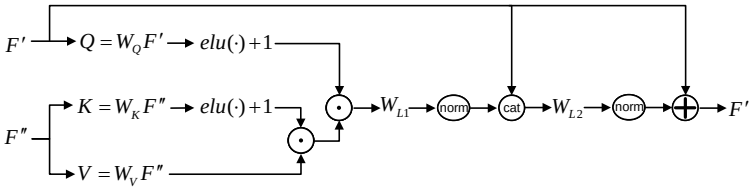

Figure 3: The self- and cross-attention architecture.

Subsequently, the feature map after adding the positional encoding is expressed as follows:

$$F_i = (f_{x,y} + p_{x,y}) \in \mathbb{F}^{C \times H \times W} \tag{10}$$

3.3.3. Self- and cross-attetion

After adding the positional encoding, local feature maps of the template and sample F_i^T and F_i^S are input into self- and cross-attention to extract global and interactive information, respectively. Consequently, dense feature maps $\hat{F}_i^T$ and $\hat{F}_i^S$ are derived.

Firstly, F_i^T, F_i^S are resized to $F_i^T, F_i^S \in \mathbb{F}^{L \times C}$, where $L = H \times W$.

Next, as shown in Figure 3, the mechanism of self- and cross-attention is depicted in the diagram. Due to the difference in inputs to self-attention and cross-attention, F' and F'' are used for representation. $Q, K, V \in \mathbb{F}^{L \times C_1}$ are computed by fully connected networks W^Q, W^K, W^V.

$$\begin{aligned} Q &= W^Q F' = \left(W^Q f_{x,y}\right) \in \mathbb{F}^{L \times C_1} \\ K &= W^K F'' = \left(W^K f_{x,y}\right) \in \mathbb{F}^{L \times C_1} \\ V &= W^V F'' = \left(W^V f_{x,y}\right) \in \mathbb{F}^{L \times C_1} \end{aligned} \tag{11}$$

Referencing the Linear Transformer[41], $Atten(Q, K, V)$ is defined as follows:

$$Atten(Q, K, V) = \phi(Q)\left(\phi(K)^\top V\right) \tag{12}$$

where $\phi(\cdot) = elu(\cdot) + 1$.

Then, the attention map $\hat{F}'$ is obtained by concatenating and applying a residual operation to Q. Before the concatenate and residual operations, there are fully connected networks W_{L1}, W_{L2} and normalization.

To compute the self-attention features of F_i^T, let $F' = F_i^T, F'' = F_i^T$. To compute the cross-attention features of F_i^T, let $F' = F_i^T, F'' = F_i^S$. Conversely, the computations for the self-attention and cross-attention features of F_i^S are performed similarly.

Finally, the dense feature maps of the template $\hat{F}_i^T$ and sample $\hat{F}_i^S$ are calculated as per equations 5 and 6. In order to endow the model with stronger representational capacity, this study adopts a strategy similar to SuperGlue[19] and LoFTR[15], stacking multiple instances of self- and cross-attention. For dense feature extraction at different scales, different numbers of computations are stacked. More stacks are used for semantic features (deep features), while fewer stacks are used for detail features (shallow features).

3.4. Feature matching

In the preceding section, the dense feature maps of the template and sample, denoted as $\hat{F}_i^T = \left(\hat{f}_a^T\right) \in \mathbb{F}^{L\times C}$ and $\hat{F}_i^S = \left(\hat{f}_b^S\right) \in \mathbb{F}^{L\times C}$, were obtained, where $\hat{f}_a^T, \hat{f}_b^S \in \mathbb{F}^C$, $a, b \in \mathcal{A}, \mathcal{B}, \mathcal{A} = \mathcal{B} = [1, 2, \ldots, L]$, and $\left(\hat{f}_a^T, \hat{f}_b^S\right)$ represents the features at position (a, b).

Firstly, the similarity of matching descriptors are express as score matrix $\mathcal{S}_i = (s_{a,b}) \in \mathbb{F}^{\mathcal{A}\times\mathcal{B}}$:

$$s_{a,b} = \left\langle \hat{f}_a^T, \hat{f}_b^S \right\rangle, \forall (a, b) \in \mathcal{A} \times \mathcal{B} \tag{13}$$

where $\langle \cdot, \cdot \rangle$ is the inner product.

Then, a dual-softmax operator [38] is applied on both dimensions of $\mathcal{S}_i$ to obtain the probability of soft mutual nearest neighbor matching. The matching probability $\mathcal{P}_i = (p_{a,b}) \in \mathbb{F}^{\mathcal{A}\times\mathcal{B}}$ is obtained by:

$$\mathcal{P}_i (a, b) = softmax (\mathcal{S}_i (a, \cdot))_b \cdot softmax (\mathcal{S}_i (\cdot, b))_a \tag{14}$$

Based on the matching probability $\mathcal{P}_i$, potential background matching features are selected by enforcing the Mutual Nearest Neighbor (MNN) criteria:

$$\mathcal{M}_i = \left\{ \left(\widetilde{a}, \widetilde{b}\right) | \forall \left(\widetilde{a}, \widetilde{b}\right) \in MNN (\mathcal{P}_i), \mathcal{P}_i \left(\widetilde{a}, \widetilde{b}\right) \geqslant \theta \right\} \tag{15}$$

where $\mathcal{M}_i = \left\{ \left(\widetilde{a}, \widetilde{b}\right) \right\}_{j=1}^{N}$ represents the matching pairs. The pseudocode for MNN can be found in Algorithm 1. Additionally, a threshold of θ is applied to filter out noise and maintain high confidence matches.

Algorithm 1 Mutual Nearest Neighbor (MNN) Algorithm

Require: Distance matrix $\mathcal{P}$, Index sets $\mathcal{A}$, $\mathcal{B}$
Ensure: Set of matching pairs $\mathcal{M}$

```
Initialize M ← ∅
for a ∈ A do
    b ← arg min_{k∈B}(P_{a,k})
    if a == arg min_{l∈A}(P_{l,b}) then
        M ← M ∪ {(a, b)}
    end if
end for
return M
```

Finally, one-to-one correspondence between the template and sample features $\mathcal{F}_i^M : M_i^S \rightarrow M_i^T$ is obtained based on $\mathcal{M}_i$, where $M_i^S = \left\{ \left(x^S, y^S\right)_j \right\}_{j=1}^{N}$ and $M_i^T = \left\{ \left(x^T, y^T\right)_j \right\}_{j=1}^{N}$. The noise-free features F_i^D is eliminated by equation 8.

3.5. Multi-feature fusion

Multi-scale fusion is an effective method for improving segmentation accuracy according to existing methodologies [42, 42, 43, 44, 45]. Through the Siamese network, five scales of features are obtained in this study. However, matching each scale would lead to significant computational overhead. In reference to Bisenet [20], fusing detailed and semantic features not only reduces computational costs but also enhances accuracy. Therefore, this study solely achieves noise-free feature maps through matching at the 1/8, 1/32 scales, while direct subtraction is applied at other scales. Ultimately, multi-scale fusion is conducted in a manner similar to U-Net as shown in Figure 2.

3.6. Loss function

In surface defect detection images, the foreground is sparse compared to the background. Therefore, this study employs Focal Loss [46] as the loss function.

Focal Loss is a loss function designed specifically to address class imbalance problem in one-stage object detection. It has proven to be effective in giving more importance to hard-to-classify instances. The Focal Loss is designed to add a modulating factor to the standard Cross Entropy criterion,

to down-weight easy examples and thus focus training on hard negatives. The Focal Loss is defined as:

$$FL(p,y) = \begin{cases} -(1-p)^{\gamma} log(p) & \text{if } y = 1 \\ -p^{\gamma} log(1-p) & \text{otherwise} \end{cases} \tag{16}$$

where p is the model's estimated probability for the class with label y, and γ is the focusing parameter that should be greater than 0. In this paper, $\gamma = 2$.

4. Experiments and Results

4.1. Experimental setup

4.1.1. Implementation details

Employing an NVIDIA GeForce RTX 4090 GPU facilitated efficient data processing, ideal for complex machine learning tasks. A PyTorch-based model was utilized, optimized via the Adam optimizer set at a learning rate of 10e-5. This arrangement ensured a balance between convergence speed and training stability. Further optimization occurred through data processing in mini-batches of eight, enabling superior GPU utilization and accelerated model updates.

4.1.2. Evaluation metrics

In this paper, we used six key metrics: Precision (Pre), Recall (Rec), F-measure (F2), mean Intersection over Union (mIoU), mean Accuracy (mACC), and Parameter size (MB).

$$Pre = \frac{TP}{TP + FP} \tag{17}$$

$$Rec = \frac{TP}{TP + FN} \tag{18}$$

$$F2 = (1 + 2^2) \cdot \frac{Pre \cdot Rec}{(2^2 \cdot Pre) + Rec} \tag{19}$$

$$mIoU = \frac{1}{N} \sum_{i=1}^{N} \frac{TP_i}{TP_i + FP_i + FN_i} \tag{20}$$

$$mACC = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FN_i} \tag{21}$$

Where TP represents the number of true positives, FP represents the number of false positives, FN represents the number of false negatives, and N represents the total number of classes.

4.1.3. Datasets description

In this paper, we focus on the challenge of background generalization for chips of surface-mounted devices, such as Optical Communication Devices (OCDs) and Printed Circuit Boards (PCBs). In these cases, background features in templates and samples exhibit spatial variations, such as shifts and rotations.

OCDs are devices that convert optical and electrical signals in Gigabit Passive Optical Networks and Optical Network Terminals. They are composed of a base, pins, and various Surface Mounted Device (SMD) components, interconnected by jump wires. The OCDs dataset [11] contains a total of 918 data sets, including 60 instances of base crushing, 27 instances of base scratches, 375 instances of component contamination, 240 instances of component breakage, and 216 instances of varying numbers of jump wires.

PCBs serve as the foundational building blocks in electronics, providing a platform that connects and supports various electronic components through conductive pathways etched from copper sheets laminated onto a non-conductive substrate. The PCBs dataset [9] consists of 340 pairs of images from a PCB manufacturer. Each pair of images includes a defective image (also referred to as an NG image) and a non-defective image (alternatively known as a template image or an OK image).

4.2. Visualization of feature matching

Figure 4 presents the Class Activation Maps (CAM) for the template and sample features. As depicted, at a 1/32 scale, BGNet accurately achieves the matching of background features between the template and the sample, focusing exclusively on the foreground defect features after corresponding subtraction. At a 1/8 scale, the detail features are highly dense, and the matching relationship is generally accurate. The corresponding subtraction retains the defect features. It should also be noted that the corresponding subtraction operation eliminates only significant features, not all background features, yet it remains quite effective.

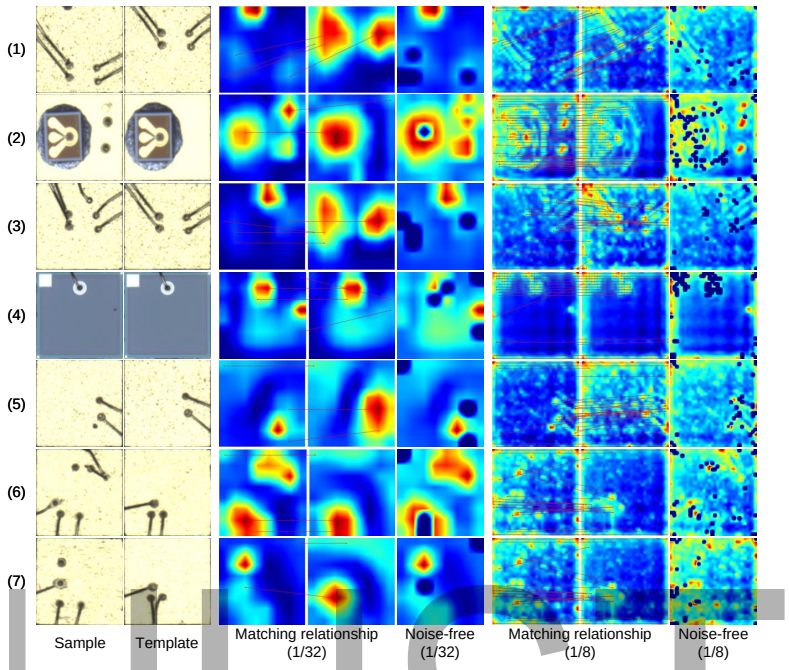

Figure 4: Visualization of feature matching results in the OCDs dataset

4.3. Ablation studies and discussion

4.3.1. Ablation experiment setting

To validate the effectiveness of the proposed BGNet, ablation experiments have been set up as follows:

S1: Concatenate the template and sample directly and input them into a U-Net-like base network, testing whether a CNN-based network inherently possesses template-sample contrast capabilities.

S2: Utilize a Siamese network in the encoding part, inputting the template and sample into a weight-shared backbone, directly subtract the features obtained from the five scales, and achieve segmentation results after feature fusion.

S3: Employ self-attention and cross-attention to further extract dense features at the 1/8 and 1/32 scales, then directly subtract and obtain segmentation results following feature fusion.

S4: Directly match the features extracted from the Siamese network at the 1/8 and 1/32 scales, subtract accordingly, and acquire segmentation results following feature fusion.

Table 1: Results of Ablation in the OCDs dataset

Modules	Baseline	Siamese	Self- and Cross-attention	Feature matching	mIoU	F2
S1	✓				0.7637	0.8683
S2	✓	✓			0.7887	0.8850
S3	✓	✓	✓		0.8071	0.8948
S4	✓	✓		✓	0.7885	0.9048
S5	✓	✓	✓	✓	0.8210	0.9101

S5: Extract dense features through self-attention and cross-attention at the 1/8 and 1/32 scales, subtract after feature matching, and obtain segmentation results following feature fusion. This represents the complete method proposed in this study.

4.3.2. Discussion of the results of the ablation experiment

The quantitative results of the ablation experiments are shown in Table 1, and the typified results are depicted in Figure 5.

Quantitatively, when the template and sample are concatenated and input into the network (S1), the network's implicit contrasting capability is found to be limited. Based on the Siamese network, subtracting corresponding features at different scales (S2) improves the mIoU by 2.50%. In S3, the mIoU increases by 4.34%. Using self- and cross-attention to extract dense features containing global and mutual information helps the model to focus on defect features and to some extent ignore background features. In S4, the mIoU increased by 2.48%, which is nearly the same as the result of the twin network in S2. This suggests that without global and mutual information, the matching algorithm contributes very little to the mIoU. However, the F2 score significantly improved, indicating a substantial increase in recall, which demonstrates that direct matching still contributes significantly to background noise shielding. In S5, the mIoU increased by 5.73%, and the F2 score was also the highest, suggesting that the dense features extracted by self-attention and cross-attention greatly assist the matching algorithm. This validates that every component of BGNet is valuable, and the most effective results are achieved when they are combined.

Qualitatively, S5 accurately focuses on the defect foreground features at scales 1/8 and 1/16. The deep blue areas, which are subtracted through

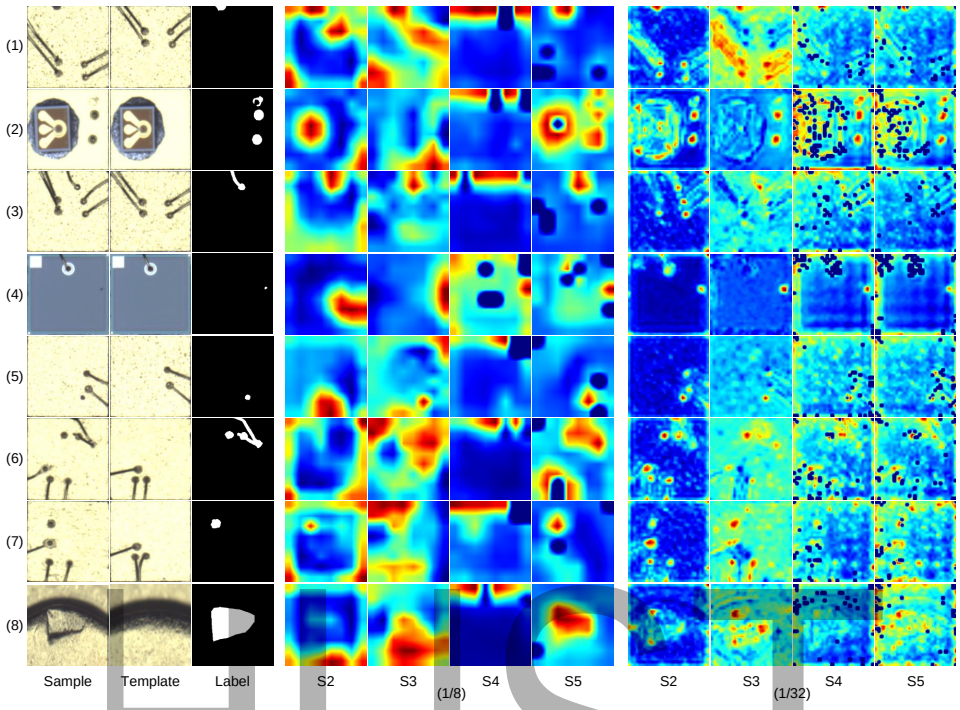

Figure 5: Visual ablation in the OCDs dataset

matching, also display spatial changes in the sample background. Moreover, compared to S2, S3 pays more attention to the defect foreground at the 1/8 scale, but there remains a significant amount of noise caused by spatial variations at the 1/32 scale. Conversely, S4 precisely eliminates noise features at the 1/32 scale, but it underperforms at the 1/8 scale.

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

The effectiveness of BGNet is demonstrated by comparing it with fifteen existing methods, including: Five classical methods that concatenate the sample and template and input them into the network. This approach exploits the network's potential to adapt to background spatial changes. Four classic semantic segmentation networks (U-Net [42], FCN [43], SegNet [44], DeepLabV3+ [47]) and a classic surface defect detection network (PGANet [45]) were selected. Four methods based on attention mechanisms, including three classical attention mechanism methods (CCNet [48], DUNet [48], DANet[49]) and a recent method based on the Transformer, Swin U-Net [50]. Two methods for foreground generalization, TGRNet [26] and PFENet[51],

Table 2: Quantitative comparison with state-of-the-arts methods in OCDs datasets

	Method	Pre	Recall	F2	mIoU
Classical methods	U-Net	0.8597	0.6926	0.7177	0.6325
	FCN	0.8831	0.7376	0.7627	0.6580
	SegNet	0.8949	0.3907	0.4403	0.3662
	DeepLabV3+	0.8295	0.7967	0.8031	0.6702
	PGANet	0.9186	0.4793	0.5300	0.4483
Attention-based methods	CCNet	0.8224	0.3875	0.4333	0.3614
	DUNet	0.8716	0.3100	0.3559	0.2942
	DANet	0.8220	0.5748	0.6116	0.5130
	Swin U-Net	0.6612	0.2569	0.2927	0.2076
Foreground generalization methods	TGRNet	0.2446	0.4770	0.4008	0.1638
	PFENet	0.1929	0.2713	0.2509	0.1233
Background generalization methods	Siamese U-Net	0.8913	0.6946	0.7267	0.6243
	DSSSNet	0.8931	0.8148	0.8293	0.7405
	GWNet	**0.9070**	0.8891	0.8926	0.8074
Ours	BGNet	0.8961	**0.9137**	**0.9101**	**0.8210**

were selected to validate that foreground generalization has limited applicability to background generalization. Three methods for background generalization, including all contrast-based background generalization methods such as Siamese-Unet [13], DSSSNet[9], and GWNet[11].

4.4.1. Comparison in OCDs dataset

In this section, we compare our approach with state-of-the-art methods from both quantitative (as depicted in Table 2) and qualitative perspectives (as illustrated in Figure 6).

Quantitatively, classic CNN-based networks exhibit limited implicit contrasting capability between templates and samples. Similarly, the performance of attention mechanism-based methods is also restricted. Foreground generalization methods do not provide positive contributions to the problem of background generalization that we study; their performance is even worse.

Among the background generalization methods, Siamese U-Net has a similar structure to U-Net. Despite the explicit feature subtraction, the performance hardly improves (mIoU is 62.43%). This demonstrates on one hand that concatenating and inputting into the network can exploit the potential of CNN-based networks to contrast templates and samples. On the other

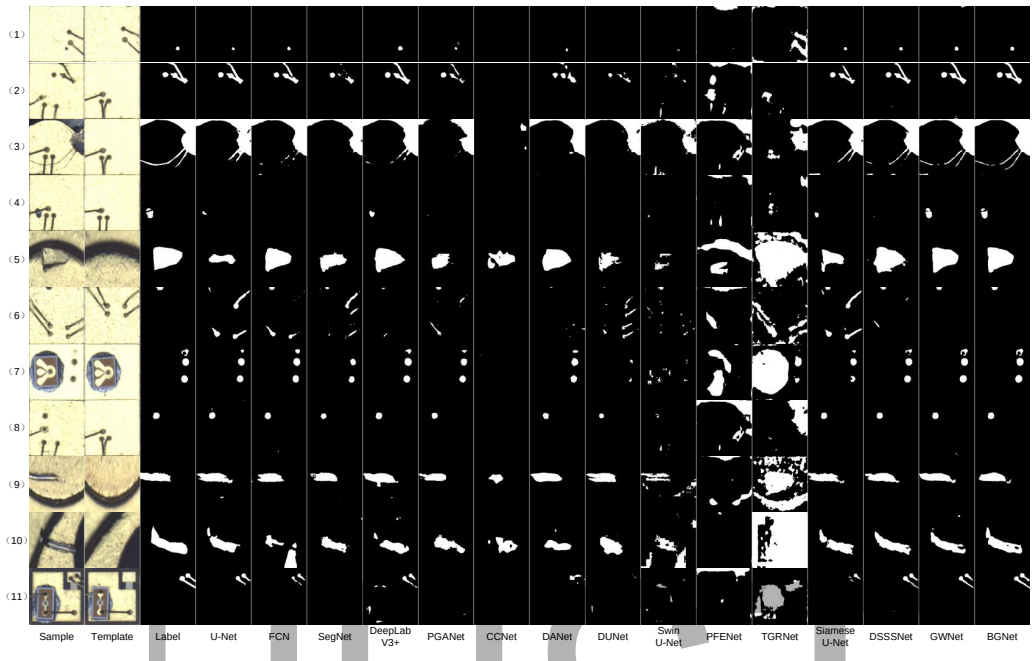

Figure 6: Visual comparison with state-of-the-arts methods in OCDs

hand, it indicates that direct subtraction cannot resolve the noise brought about by spatial changes in background features.

DSSSNet adopts a measure similar to max pooling, achieving some degree of noise elimination due to spatial variations, leading to a significant improvement in mIoU (74.05%). This substantiates that eliminating noise caused by background changes is a key measure to achieve background generalization.

GWNet employs an attention mechanism and based on the location-independent characteristics of the attention mechanism, it further enhances the model's ability to eliminate background noise, raising the mIoU to 80.74%.

The feature matching-based method proposed in this paper explicitly accomplishes the elimination of background noise and further raises the mIoU to 82.10%, surpassing state-of-the-art methods.

Qualitatively, classical methods exhibit significant false negatives and false positives. False positives primarily occur in areas with spatial changes in background features, such as the gold wire in row (6). False negatives mainly occur in areas where foreground defect features overlap with spatially varying background features, such as rows (3) and (4). Attention-based methods, such as CCNet and Swin U-Net, perform poorly, with numerous false

Table 3: Quantitative comparison with state-of-the-arts methods in PCBs datasets

	Method	mIoU	mACC	Params (MB)
Classical methods	U-Net	0.5981	0.9046	7.86
	FCN	0.4985	0.8203	15.32
	SegNet	0.7864	0.9974	40.47
	DeepLabV3+	0.7094	0.9351	32.98
	PGANet	0.7894	0.9975	51.41
Attention-based methods	CCNet	0.4786	0.9087	67.70
	DUNet	0.7513	0.9126	31.48
	DANet	0.7243	0.9003	49.63
	Swin-U-Net	0.7719	0.9972	27.16
Foreground generalization methods	TGRNet	0.7068	0.8988	32.12
	PFENet	0.7214	0.9105	30.25
Background generalization methods	Siamese U-Net	0.7837	0.9954	**7.85**
	DSSSNet	0.7634	0.9678	33.60
	GWNet	0.8243	0.9978	26.54
Ours	BGNet	**0.8393**	**0.9996**	47.70

negatives and severe false positives, respectively. Although these methods have achieved some contrasting ability between the template and the sample, their ability to distinguish between the background and the foreground is limited.Foreground generalization methods, such as PEFNet and TGRNet, perform the worst and are almost incapable of correctly detecting defects.

Among background generalization methods, DSSSNet reduces the incidence of false positives compared to Siamese U-Net and greatly improves noise removal ability, such as in row (6). However, it also exhibits some false negatives, such as in row (4). GWNet shows promising detection results but lacks accuracy in detecting details compared to BGNet, such as in rows (4) and (10). It also exhibits some minor noise, such as in rows (1), (3), (8), and (10).

4.4.2. Comparison in PCBs dataset

In order to further validate the effectiveness of BGNet, we added a multi-class dataset, the PCBs Dataset. Quantitative results are presented in Table 3, and qualitative results are shown in Figure 7.

Quantitatively, similar to the results on the OCDs dataset, conven-

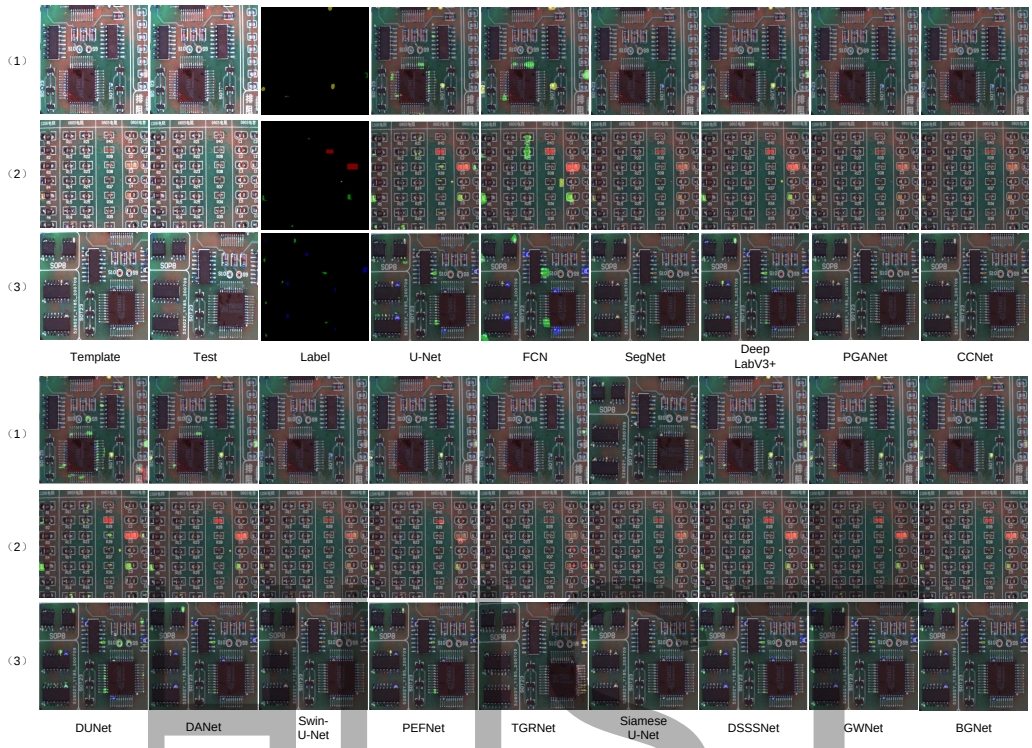

Figure 7: Visual comparison with state-of-the-arts methods in PCBs

tional methods based on CNN and attention mechanisms demonstrated limited effectiveness, while foreground generalization methods performed the poorest. Among the background generalization methods, Siamese U-Net and GWNet outperformed DSSSNet. This may be attributed to the fact that methods similar to max pooling have predefined pooling ranges, which necessitate adaptation to the scale of defects and background features. Such predefined spatial variation adaptation methods have inherent limitations. BGNet achieved the best results among all methods. Although the number of parameters in BGNet has increased, it remains within an acceptable range compared to existing methods.

Qualitatively, it can be observed that in the PCBs dataset, the background components are more numerous and densely distributed, and the spatial variation of background features between samples and templates is relatively small. This is consistent with the small difference in mIoU of various methods shown in Table 3. In Figure 6, the classical methods based

on CNN and those based on attention mechanisms have a lot of false positives. Foreground generalization methods have a lot of false negatives. The background generalization methods generally performed well, and BGNet achieved a very high accuracy.

5. Discussion and Conclusion

5.1. Discussion

The comparison of template and sample is an effective and widely used method in defect detection models. This paper addresses the issue of background generalization, specifically the spatial variation of background features, which introduces noise into the comparison. Based on our ideas and experimental results, we discuss the following points:

1) Our experimental results show that CNN networks do not exhibit spatial invariance properties. Existing research also indicates that CNN networks are spatially equivariant. The pooling operation can provide CNN networks with limited spatial invariance properties, as demonstrated by the improved performance of DSSSNet on OCDs. However, due to the definition of the pooling operation, its receptive field is fixed, resulting in poor performance of DSSSNet on PCBs.

2) The self-attention and cross-attention mechanisms in Transformers have greater equivariance properties for spatial changes in background features due to their ability to capture global and interactive information. However, the underlying principle of this equivariance remains unclear. In GWNet, position encoding was not performed prior to calculating self-attention. This leverages the location-independent nature of self-attention to reduce the impact of spatial variation noise on the results. BGNet, on the other hand, incorporated position encoding, which also helped mitigate the effects of spatial variation noise.

3) When subtracting feature matches with background variation, it is not necessary to subtract all backgrounds individually. Instead, subtracting only significant features allows the network to eliminate noise caused by spatial variations.

5.2. Conclusion

In this study, we introduce a novel Background Generalization Network (BGNet) that leverages feature matching to achieve state-of-the-art results.

Our network employs self-attention and cross-attention mechanisms to extract dense features containing global and interactive information. Feature matching is accomplished using the Mutual Nearest Neighbor (MNN) algorithm, and subtraction is performed based on the matching relationship to explicitly eliminate spatially variant background features. Our proposed method demonstrates exceptional performance on both the OCDs and PCBs datasets. Future work will focus on exploring the mathematical principles underlying spatial variations and designing networks based on matrix translation, rotation, and affine transformation to further elucidate the mechanisms governing spatial variations in background features.

References

[1] S.-H. Chen, C.-C. Tsai, Smd led chips defect detection using a yolov3-dense model, Advanced Engineering Informatics 47 (2021) 101255. doi:https://doi.org/10.1016/j.aei.2021.101255.
URL https://www.sciencedirect.com/science/article/pii/S1474034621000100

[2] Q. Lu, J. Lin, L. Luo, Y. Zhang, W. Zhu, A supervised approach for automated surface defect detection in ceramic tile quality control, Advanced Engineering Informatics 53 (2022) 101692. doi:https://doi.org/10.1016/j.aei.2022.101692.
URL https://www.sciencedirect.com/science/article/pii/S1474034622001525

[3] D. Li, Q. Xie, X. Gong, Z. Yu, J. Xu, Y. Sun, J. Wang, Automatic defect detection of metro tunnel surfaces using a vision-based inspection system, Advanced Engineering Informatics 47 (2021) 101206. doi:https://doi.org/10.1016/j.aei.2020.101206.
URL https://www.sciencedirect.com/science/article/pii/S1474034620301750

[4] Z. Pourkaramdel, S. Fekri-Ershad, L. Nanni, Fabric defect detection based on completed local quartet patterns and majority decision algorithm, Expert Systems with Applications 198 (2022) 116827. doi:https://doi.org/10.1016/j.eswa.2022.116827.
URL https://www.sciencedirect.com/science/article/pii/S0957417422002834

[5] W. Li, B. Li, S. Niu, Z. Wang, B. Liu, T. Niu, Selecting informative data for defect segmentation from imbalanced datasets via active learning, Advanced Engineering Informatics 56 (2023) 101933. doi:https://doi.org/10.1016/j.aei.2023.101933.
URL https://www.sciencedirect.com/science/article/pii/S1474034623000617

[6] W. Zhu, H. Zhang, C. Zhang, X. Zhu, Z. Guan, J. Jia, Surface defect detection and classification of steel using an efficient swin transformer, Advanced Engineering Informatics 57 (2023) 102061. doi:https://doi.org/10.1016/j.aei.2023.102061.
URL https://www.sciencedirect.com/science/article/pii/S1474034623001891

[7] H. Shang, C. Sun, J. Liu, X. Chen, R. Yan, Defect-aware transformer network for intelligent visual surface defect detection, Advanced Engineering Informatics 55 (2023) 101882. doi:https://doi.org/10.1016/j.aei.2023.101882.
URL https://www.sciencedirect.com/science/article/pii/S1474034623000101

[8] T. Liu, Z. He, Z. Lin, G.-Z. Cao, W. Su, S. Xie, An adaptive image segmentation network for surface defect detection, IEEE Transactions on Neural Networks and Learning Systems (2022) 1–14doi:10.1109/TNNLS.2022.3230426.

[9] Z. Ling, A. Zhang, D. Ma, Y. Shi, H. Wen, Deep siamese semantic segmentation network for pcb welding defect detection, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–11. doi:10.1109/TIM.2022.3154814.

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

[11] T. Niu, Z. Xie, J. Zhang, L. Tang, B. Li, H. Wang, A generalized well neural network for surface defect segmentation in optical communication devices via template-testing comparison, Computers in Industry 151 (2023) 103978. doi:https://doi.org/10.1016/j.compind.2023.103978.

URL https://www.sciencedirect.com/science/article/pii/S0166361523001288

[12] M. Jaderberg, K. Simonyan, A. Zisserman, et al., Spatial transformer networks, Advances in neural information processing systems 28 (2015).

[13] D. Kwon, J. Ahn, J. Kim, I. Choi, S. Jeong, Y.-S. Lee, J. Park, M. Lee, Siamese u-net with healthy template for accurate segmentation of intracranial hemorrhage, in: D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, A. Khan (Eds.), Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, Springer International Publishing, Cham, 2019, pp. 848–855.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[15] J. Sun, Z. Shen, Y. Wang, H. Bao, X. Zhou, Loftr: Detector-free local feature matching with transformers, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 8922–8931.

[16] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, C.-L. Tai, Transfusion: Robust lidar-camera fusion for 3d object detection with transformers, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 1090–1099.

[17] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, D. Tao, Gmflow: Learning optical flow via global matching, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 8121–8130.

[18] S. Zhu, X. Liu, Pmatch: Paired masked image modeling for dense geometric matching, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 21909–21918.

[19] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, Superglue: Learning feature matching with graph neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4938–4947.

[20] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, N. Sang, Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation, International Journal of Computer Vision 129 (2021) 3051–3068.

[21] D.-M. Tsai, P.-H. Jen, Autoencoder-based anomaly detection for surface defect inspection, Advanced Engineering Informatics 48 (2021) 101272. doi:https://doi.org/10.1016/j.aei.2021.101272.
URL https://www.sciencedirect.com/science/article/pii/S1474034621000276

[22] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, S. J. Lee, Automated defect inspection system for metal surfaces based on deep learning and data augmentation, Journal of Manufacturing Systems 55 (2020) 317–324.

[23] W. Xiao, K. Song, J. Liu, Y. Yan, Graph embedding and optimal transport for few-shot classification of metal surface defect, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–10.

[24] Y. Song, Z. Liu, S. Ling, R. Tang, G. Duan, J. Tan, Coarse-to-fine few-shot defect recognition with dynamic weighting and joint metric, IEEE Transactions on Instrumentation and Measurement 71 (2022) 1–10. doi:10.1109/TIM.2022.3193204.

[25] W. Zhao, K. Song, Y. Wang, S. Liang, Y. Yan, Fanet: Feature-aware network for few shot classification of strip steel surface defects, Measurement 208 (2023) 112446.

[26] Y. Bao, K. Song, J. Liu, Y. Wang, Y. Yan, H. Yu, X. Li, Triplet-graph reasoning network for few-shot metal generic surface defect segmentation, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–11.

[27] D. Shan, Y. Zhang, S. Coleman, D. Kerr, S. Liu, Z. Hu, Unseen-material few-shot defect segmentation with optimal bilateral feature transport network, IEEE Transactions on Industrial Informatics 19 (7) (2023) 8072–8082. doi:10.1109/TII.2022.3216900.

[28] X. Shi, S. Zhang, M. Cheng, L. He, X. Tang, Z. Cui, Few-shot semantic segmentation for industrial defect recognition, Computers in Industry

148 (2023) 103901. doi:https://doi.org/10.1016/j.compind.2023.103901.
URL https://www.sciencedirect.com/science/article/pii/S0166361523000519

[29] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2004) 91–110.

[30] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: An efficient alternative to sift or surf, in: 2011 International Conference on Computer Vision, 2011, pp. 2564–2571. doi:10.1109/ICCV.2011.6126544.

[31] M. Calonder, V. Lepetit, C. Strecha, P. Fua, Brief: Binary robust independent elementary features, in: Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11, Springer, 2010, pp. 778–792.

[32] K. M. Yi, E. Trulls, V. Lepetit, P. Fua, Lift: Learned invariant feature transform, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14, Springer, 2016, pp. 467–483.

[33] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2018, pp. 224–236.

[34] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, T. Sattler, D2-net: A trainable cnn for joint detection and description of local features, arXiv preprint arXiv:1905.03561 (2019).

[35] Y. Ono, E. Trulls, P. Fua, K. M. Yi, Lf-net: Learning local features from images, Advances in neural information processing systems 31 (2018).

[36] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, M. Humenberger, R2d2: repeatable and reliable detector and descriptor, arXiv preprint arXiv:1906.06195 (2019).

[37] C. Liu, J. Yuen, A. Torralba, Sift flow: Dense correspondence across scenes and its applications, IEEE transactions on pattern analysis and machine intelligence 33 (5) (2010) 978–994.

[38] I. Rocco, M. Cimpoi, R. Arandjelović, A. Torii, T. Pajdla, J. Sivic, Neighbourhood consensus networks, Advances in neural information processing systems 31 (2018).

[39] I. Rocco, R. Arandjelović, J. Sivic, Efficient neighbourhood consensus networks via submanifold sparse convolutions, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16, Springer, 2020, pp. 605–621.

[40] X. Li, K. Han, S. Li, V. Prisacariu, Dual-resolution correspondence networks, Advances in Neural Information Processing Systems 33 (2020) 17346–17357.

[41] A. Katharopoulos, A. Vyas, N. Pappas, F. Fleuret, Transformers are rnns: Fast autoregressive transformers with linear attention, in: International conference on machine learning, PMLR, 2020, pp. 5156–5165.

[42] O. Ronneberger, P. Fischer, 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.

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

[44] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE transactions on pattern analysis and machine intelligence 39 (12) (2017) 2481–2495.

[45] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, Q. Meng, Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection, IEEE Transactions on Industrial Informatics 16 (12) (2019) 7448–7458.

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

[47] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, 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.

[48] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, Ccnet: Criss-cross attention for semantic segmentation, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 603–612.

[49] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, R. Su, Dunet: A deformable network for retinal vessel segmentation, Knowledge-Based Systems 178 (2019) 149–162.

[50] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, M. Wang, Swin-unet: Unet-like pure transformer for medical image segmentation, in: European conference on computer vision, Springer, 2022, pp. 205–218.

[51] Z. Tian, H. Zhao, M. Shu, Z. Yang, R. Li, J. Jia, Prior guided feature enrichment network for few-shot segmentation, IEEE transactions on pattern analysis and machine intelligence 44 (2) (2020) 1050–1065.