Bidirectional Feature Pyramid Siamese Anomaly Detection Network with Cellular Anomaly Generation for Container Marking

Yikui Zhai, *Senior Member, IEEE*, Wenfeng Pan, *Graduate Student Member, IEEE*, Yanyang Liang, Hufei Zhu, Zhihao Long, Pasquale Coscia, *Senior Member, IEEE*, Angelo Genovese, *Senior Member, IEEE*, Vincenzo Piuri, *Fellow, IEEE*, Fabio Scotti, *Senior Member, IEEE*

***Abstract*—Container marking anomaly detection (CMAD) aims to identify markings that deviate from a standard reference image. Currently, the manual execution of CMAD is inefficient and susceptible to errors. Existing machine vision methods are limited to detecting text-based or specific types of markings, rendering type-agnostic CMAD unattainable. To address this limitation, we propose an innovative framework for CMAD, named Bidirectional Feature Pyramid Siamese Anomaly Detection Network (BiSiNet). BiSiNet consists of the Siamese Encoder (SE), the Bidirectional Feature Pyramid Decoder (BiD), and the Triplet Classifier Prediction Head (TCPH). SE extracts the difference feature pyramid between the target and reference images. BiD decodes the difference feature pyramid in both top-down and bottom-up paths, effectively solving the problem of global information loss during decoding in traditional single-path decoder. TCPH utilizes both local and global features to predict outcomes at both the pixel and patch levels, integrating these results to resolve inconsistencies in predictions. BiSiNet not only possesses strong resistance to disturbances such as reflections and shadows but also has accurate localization capabilities. Due to the limited diversity and the difficulty in obtaining anomalous samples, we designed an innovative Cellular Anomaly Generation Strategy (CAGS). CAGS employs Cellular Noise to generate marking masks, creating realistic pseudo marking images. Through innovative design, it produces four types of samples, thereby enhancing sample diversity and improving training generalization. Extensive experiments using our established ContainerMAD dataset have demonstrated that BiSiNet and CAGS outperform other methods in both container marking anomaly detection and localization. Additionally, favorable outcomes have been achieved on the MvtecAD dataset.**

This study was funded by Development of a Container Intelligent Panoramic Trademark Quality Inspection System Based on Machine Vision (HX22105), Guangdong Basic and Applied Basic Research Foundation (No.2021A1515011576), Guangdong Science and Technology Planning Project (No.2021A0505030080, No.2021A0505060011), Guangdong Higher Education Innovation and Strengthening School Project (No.2020ZDZX3031, No.2022ZDZX1032, No.2023ZDZX1029), Guangdong Jiangmen Science and Technology Research Project (No.2220002000246), Wuyi University Innovation and Entrepreneurship Foundation (2023040300000156). (*Corresponding author: Yikui Zhai.*)

Yikui Zhai, Wenfeng Pan, Yanyang Liang, Hufei Zhu, and Zhihao Long are with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, 529020, China (e-mail: yikuizhai@163.com; wenfengpan97@163.com; liangyanyang@163.com; zhuhufei@aliyun.com; longzhihao@wyu.edu.cn).

Pasquale Coscia, Angelo Genovese, Vincenzo Piuri, and Fabio Scotti are with the Department of Computer Science, Università Degli Studi di Milano, 20133, Milan, Italy (e-mail: pasquale.coscia@unimi.it; angelo.genovese@unimi.it; vincenzo.piuri@unimi.it; fabio.scotti@unimi.it)

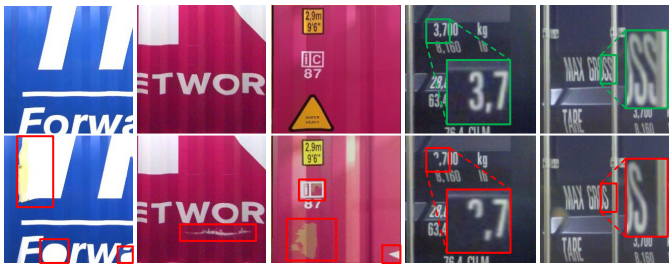


Fig. 1. Examples of container marking anomalies. The top row displays anomaly-free reference images, while the bottom row illustrates anomalous images. The red-bordered area denotes the anomalous region.

***Index Terms*—Shipping container, container marking anomaly detection, anomaly generation, bidirectional decoder.**

I. Introduction

CONTAINER marking anomaly detection (CMAD) constitutes a critical phase in the container production process, aiming to identify deviations from standard marking reference images. As depicted in Fig. 1, typical anomalies in container markings encompass incorrect, missing, or damaged markings. These issues can result in significant consequences such as shipping delays, misrouting of goods, damage to the manufacturing company's reputation, and increased safety risks. Accurate detection of container marking anomalies is therefore paramount. Currently, factories rely on manual inspection for detecting container marking anomalies, which involves multiple inspectors or inspection rounds around containers. This approach suffers from high labor intensity, low efficiency, potential for errors, and certain safety risks.

In recent years, deep learning has rapidly advanced and has been widely applied across various industrial sectors, including the container industry [1]–[8]. Currently, deep learning-based methods for container marking anomaly detection primarily utilize Optical Character Recognition (OCR) technology [1], [3]. As container markings often contain textual information, researchers employ text detection and recognition technologies [9]–[11] to identify these markings and compare them against production standards to detect anomalies. However, the robust generalization capability of deep learning methods can result in correctly recognizing original information

even when markings are partially damaged, thereby failing to detect damaged text anomalies [12], [13]. Furthermore, besides textual markings, there are numerous non-textual container markings that OCR-based methods cannot handle. Object detection-based approaches address container marking anomaly detection by detecting both the container marking and any defects [7], [8]. However, due to the wide variety of container markings and the presence of unidentified categories, it is impractical to collect data on all container markings for training, limiting the applicability of object detection-based methods to unseen markings in the training set. Consequently, OCR-based and object detection-based methods are unable to achieve genuine type-agnostic anomaly detection for container markings. Moreover, deep learning-based image anomaly detection methods typically employ unsupervised or semi-supervised learning, focusing on normal samples or pseudo-anomalous samples, and utilize reconstruction-based [14]–[16] or embedding-based [17]–[19] techniques to identify potential anomalies. While both reconstruction-based and embedding-based methods have demonstrated good performance in image anomaly detection across various industrial products, previous applications have typically involved training and testing sets with surface features from a unified distribution. In the context of container marking anomaly detection, factories produce new containers, leading to scenarios where container markings not encountered during training may appear. Consequently, existing anomaly detection methods face significant challenges when applied directly to container markings.

Recent research has investigated the use of reference-based methods for industrial anomaly detection [20], [21]. Although existing reference-based methods predominantly focus on individual categories or visually consistent industrial products, they offer effective solutions for container marking anomaly detection, particularly when confronted with unrecognized markings in the test dataset. In a production batch, there are often hundreds or even thousands of containers of the same type. Therefore, by manually confirming one anomaly-free container as a reference, standard marking reference images can be provided for the rest of the batch. Consequently, CMAD can be achieved by employing Siamese networks to detect deviations from the standard marking reference on the target image. Furthermore, we observed that current encoder-decoder networks predominantly use single-path decoders, which sequentially decode high-dimensional encoder features into low-dimensional features to generate predictions. However, such decoders gradually lose global information during decoding, leading to ineffective handling of disruptive variations such as reflections and misalignments, where local features dominate the prediction results. To tackle the aforementioned challenge, we propose an innovative framework for CMAD, named Bidirectional Feature Pyramid Siamese Anomaly Detection Network (BiSiNet). BiSiNet enhances anomaly detection by integrating top-down and bottom-up decoding to preserve both local and global context, crucial for robust anomaly identification in factory environments.

Additionally, the scarcity of anomalous data and the diversity of container markings necessitate advanced techniques for anomaly generation. Previous methods [15], [22], such as using random Perlin Noise [23] to create anomaly masks, lack structured realism in simulating container markings. Alternatively, employing random rectangular areas for masking is overly simplistic and does not effectively support training [24]. Therefore, we propose the Cellular Anomaly Generation Strategy (CAGS), which utilizes Cellular Noise [25] to generate structured marking masks resembling authentic container markings. CAGS facilitates reference-based CMAD by creating training samples comprising reference images, target images, and pixel-level anomaly masks, thereby enhancing model training effectiveness and deployment robustness.

Our main contributions are summarized as follows:

1) An inspection system for CMAD is proposed, which is implemented on a container production line. Furthermore, a procedural flow for reference-based CMAD during container production is designed.
2) A novel framework for CMAD, named BiSiNet, is proposed. BiSiNet utilizes bidirectional dual-path decoding on the feature pyramid to derive local and global features. Subsequently, these features undergo pixel-level and patch-level predictions, which are then fused. The bidirectional dual-path decoding structure of BiSiNet addresses the loss of global information typically encountered in traditional single-path decoders. BiSiNet not only possesses strong resistance to disturbances such as reflections and shadows but also has accurate localization capabilities.
3) An anomaly generation strategy named CAGS is proposed. CAGS generates four types of samples, enhancing training diversity and robustness. CAGS utilizes Cellular Noise-based masks to produce pseudo container markings that are more regular and structured, resembling real markings. This method notably enhances the network's capacity to learn precise decision boundaries and strengthens the generalization capability of semi-supervised training.
4) Extensive experiments were conducted on the ContainerMAD dataset we constructed. The results demonstrate that the proposed BiSiNet and CAGS outperform other methods in container marking anomaly detection and localization. Furthermore, experiments on the MVTecAD dataset validate BiSiNet's generalization capabilities and effectiveness in detecting surface anomalies across diverse industrial products.

II. Related Work

A. Deep Learning Applications in Container Industry

Deep learning has profoundly impacted various sectors, including the container industry. Wang et al. [5] used a transfer learning-based Convolutional Neural Network to detect nine types of container damage, significantly improving detection efficiency and accuracy. Bahrami et al. [2] introduced an automated framework for container seal detection, enhancing performance with attention mechanisms and long-term memory modules. Additionally, Bahrami et al. [6] applied neural networks to detect container corrosion.

Container codes serve as unique identifiers and are crucial pieces of information on containers, highlighting the importance of their detection and recognition. Zhang et al. [1] introduced an adaptive approach for localizing and recognizing container codes, addressing interference issues from various noises encountered by traditional methods. Xu et al. [3] went beyond container code localization and recognition by introducing a large-scale container scene text dataset, which covers various types of text on containers. Using this dataset, they trained advanced networks to achieve high-performance localization and recognition of general container text. Despite these advances, there is currently no dedicated research on CMAD.

B. Deep Learning-Based Image Anomaly Detection Methods

Reconstruction-based methods restore anomalous images to their normal counterparts and detect anomalies based on reconstruction errors. These methods are widely used in surface anomaly detection across industries and are valued for their robustness and effectiveness. Tsai et al. [14] enhanced convolutional autoencoders (CAE) by training only on normal samples and introducing regularization to improve feature distribution, enabling clear separation of defect samples during evaluation. Li et al. [26] integrated the L1 loss and structural similarity loss functions to enhance adversarial autoencoder performance, introducing a secondary reconstruction method that significantly improved image quality. Guo et al. [16] introduced bottleneck compression and template-guided methods to mitigate information loss, effectively restoring anomalous features. Despite their effectiveness in identifying discrepancies between reconstructed and original data, these methods may struggle with anomalies closely resembling normal conditions.

Embedding-based methods use deep neural networks pretrained on extensive datasets to extract features, which are then embedded into a compact representation space for anomaly detection and localization. Schirrmeister et al. [27] analyze anomaly detection in deep generative neural networks, highlighting the influence of earlier stage contributions. Cohen et al . [17] synthesized the segmentation map by aligning pixel-level anomalies with normal features using feature pyramid matching, addressing the limitation of traditional KNN in obtaining anomaly segmentation maps. Roth et al. [18] propose PatchCore to tackle the cold-start problem in industrial anomaly detection using a feature memory bank, achieving efficient recognition. Hyun et al. [19] introduce ReConPatch, leveraging pretrained models and soft-guided metric learning for effective anomaly representation and memory bank construction, demonstrating high performance.

Anomaly generation-based methods involve adding artificially generated anomalies to normal images to train deep learning models. Zavrtanik et al. [15] used binarized Perlin Noise to create anomaly maps. They blended out-of-distribution images into normal ones based on these maps, generating pseudo anomalous images for training reconstruction and discrimination networks. This approach enhances models' ability in anomaly detection and localization. Yang et al. [22], building on DRAEM [15], partitioned normal images into grids and rearranged them randomly to insert structural anomalous images. Li et al. [28] proposed CutPaste, creating local irregularity anomalies by cutting and pasting small rectangular regions within images. Chen et al. [29] synthesized artificial anomaly images by integrating real anomalies, augmented through data enhancement, onto normal images. This method allowed them to generate a substantial amount of abnormal data from a limited quantity of real anomaly data.

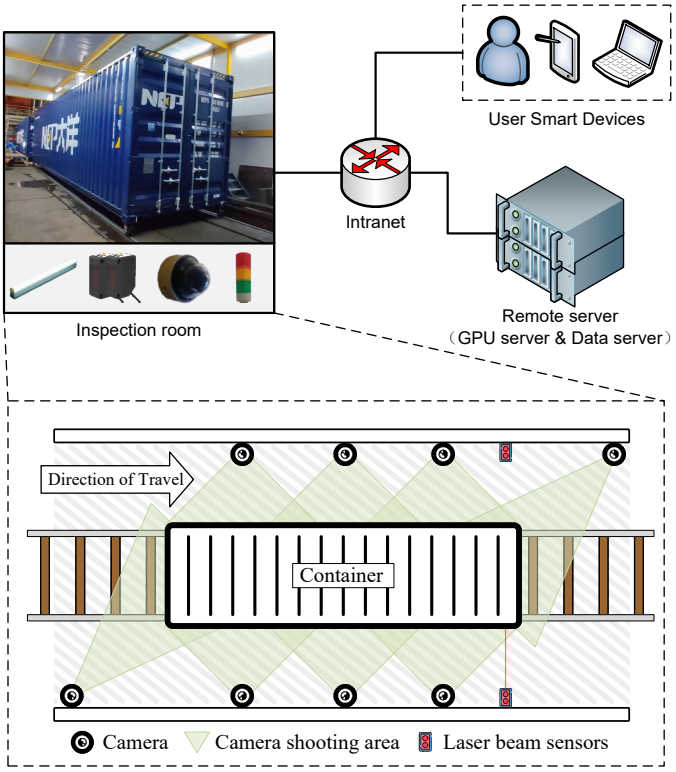


Fig. 2. Hardware architecture of CMAD system.

III. Container Marking Anomaly Detection System

A. Components of the CMAD System

Our CMAD system primarily consists of an on-site semi-enclosed inspection room, a remote server, and user smart devices, as shown in Fig. 2. The on-site semi-enclosed inspection room is equipped with multiple wide-angle cameras, laser sensors, lights, audible and visual alarms. The arrangement of the wide-angle cameras and laser sensors is illustrated in Fig. 2 (bottom). Individual cameras capture images of the front and rear of the container, while multiple cameras are used to capture images of the sides due to the container's considerable length. When the container reaches the designated position, it blocks the laser beam emitted by the emitter, preventing the receiver from detecting the signal, thereby confirming the container's arrival. Once a container enters, the wide-angle camera captures images. Given the complexity of the production site environment, we have deployed the GPU server and data center on the remote server. The network devices in the inspection room are connected to the remote server via a network switch. Staff can access the CMAD system network

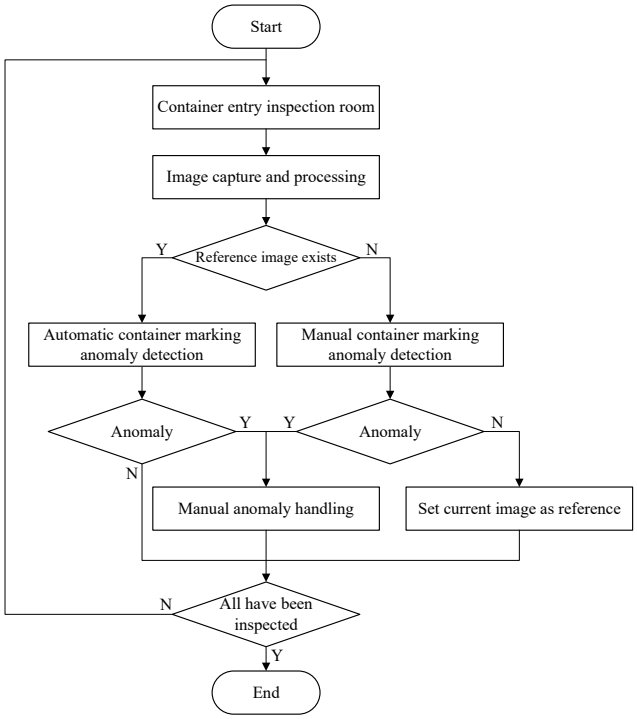


Fig. 3. Flowchart for single production batch container marking anomaly detection.

through smart devices such as computers and mobile phones to view detection data and initiate the detection system.

B. Workflow of the CMAD System

The CMAD system begins when a worker initiates anomaly detection for container marking. As a container enters the detection room, multiple wide-angle cameras capture and process images of the container. The system then checks whether there exists a reference image for the current batch. If no reference image is available, the worker conducts a manual inspection. If anomalies are detected, corrective actions are taken, and the process moves to the next container. Conversely, if no anomalies are found, the container is marked as anomaly-free, and its image is saved as the new reference for future comparisons. A batch of containers contains multiple reference images, each corresponding to a specific camera.

When a reference image exists, the CMAD system utilizes BiSiNet, our proposed CMAD framework, to compare the current container image with the reference. If anomalies are detected, appropriate actions are taken before proceeding to the next container in the batch. This iterative process continues until all containers in the production batch have been inspected, ensuring thorough anomaly detection and quality assurance. The detection flowchart of the CMAD system for a single batch of container production is shown in Fig. 3.

C. Image Processing and Anomaly Testing

After capturing the images, the system processes them before testing. Since the front and rear ends of the containers cannot be captured head-on, homography transformations are

Fig. 4. Homography transformation for the front and rear images of the container and background cropping for the side image.

applied to these images, as shown in Fig. 4 (left). Additionally, the side images require cropping of the background at the top and bottom edges, as illustrated in Fig. 4 (right). The container's position in the testing area remains nearly constant with each parking, ensuring that the relative positions of the cameras and container are also consistent. Consequently, the parameters for the homography transformation and cropping operations need to be set only once in advance.

During testing, the target image is aligned with the reference image using traditional image alignment methods. Given the large size of a single complete image, we employ a sliding window method to sample N pairs of small image patches from both the reference and target images, which are then processed separately by BiSiNet for detection. This produces N anomaly score prediction maps, which are subsequently stitched together to create a complete anomaly score prediction map. Finally, the anomaly score prediction map is binarized using a threshold value a, resulting in an anomaly segmentation map. In practical applications, the threshold a is typically set to 0.5.

IV. Proposed BiSiNet

In recent years, despite significant advancements in methods for image anomaly detection, research on reference-based anomaly detection remains limited. Most reference-based networks follow the Siamese encoder and single-path decoder [30], [31]. Networks using single-path decoder framework gradually lose global information during decoding, making them susceptible to local interference and reducing their performance. To tackle these challenges, we propose a CMAD framework named BiSiNet, consisting of three parts: SE, BiD, and TCPH. The overall framework of BiSiNet is illustrated in Fig. 5.

A. Siamese Encoder

The primary function of the SE is to extract the difference feature pyramid between the reference and target images. As illustrated in Fig. 5, we employ a Siamese ResNet18 [32] with shared weights for this purpose. To tailor ResNet18 for the CMAD task, we removed its classification head, which includes the global average pooling layer and the final fully connected layer. In the original classification head of ResNet18, the global average pooling layer produces a global feature vector but reduces spatial resolution. Given that preserving spatial information is crucial for anomaly detection and localization, we retain only the backbone network to

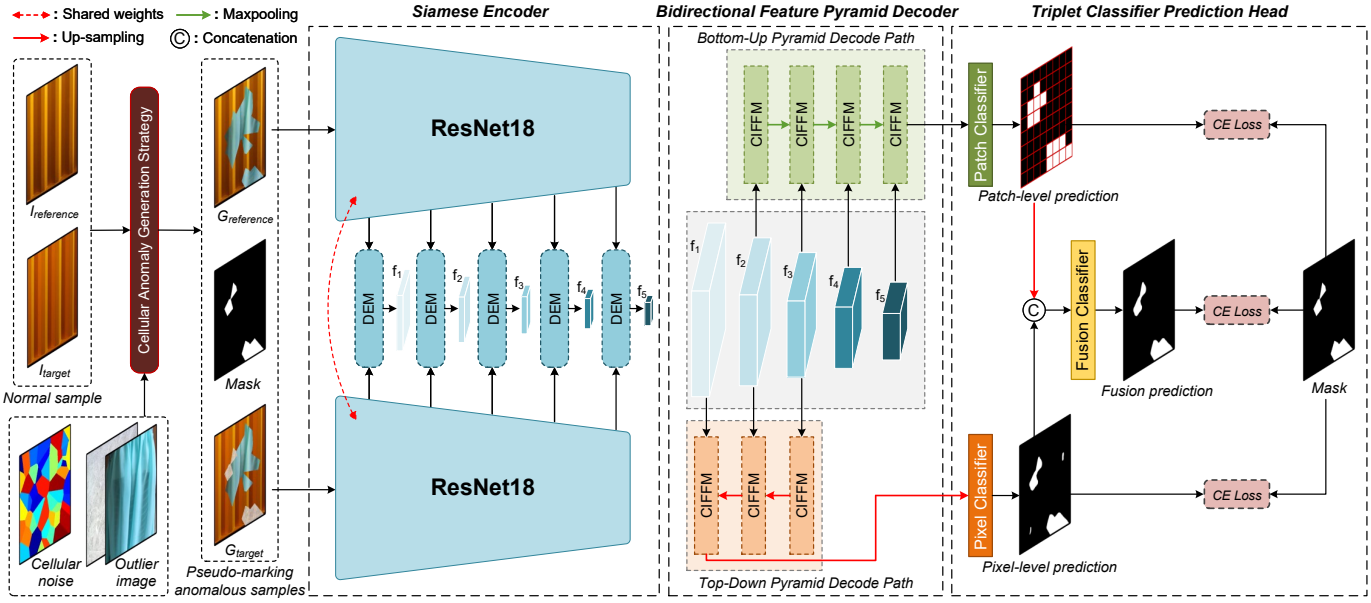


Fig. 5. Overview of proposed BiSiNet. BiSiNet consists of SE, BiD, and TCPH. SE is used to extract the difference feature pyramid between the reference and target images. BiD decodes the difference feature pyramid with bidirectional dual pathways to obtain local and global features. Finally, TCPH predicts and fuses these features. During the training phase, normal samples are initially processed by the CAGS to generate artificial anomalous samples, which are subsequently used for training. During the testing phase, samples are input directly into BiSiNet without undergoing CAGS processing.

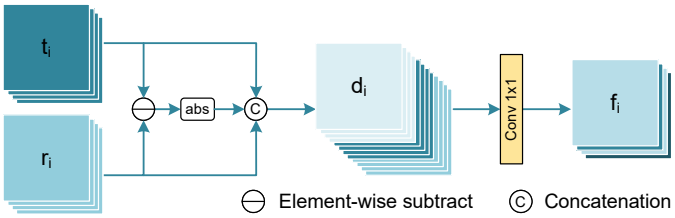


Fig. 6. Structure of DEM. DEM enhances the difference feature by concatenating the two input feature maps with their absolute difference.

generate high-resolution feature maps. When given a pair of reference and target images, the ResNet18 processes them through its five convolutional layers to produce feature maps at five different scales. These feature maps are denoted as $\{r_i \mid i = 1, 2, 3, 4, 5\}$ for the reference image and $\{t_i \mid i = 1, 2, 3, 4, 5\}$ for the target image.

Furthermore, we introduce Difference Enhancement Module (DEM) from WNet [33] to extract the difference feature pyramid DFP. DFP is derived by passing feature maps at each level through DEM. The process of extracting DFP can be formulated as follows:

$$DFP = \{f_i = \text{DEM}(r_i, t_i) \mid i = 1, 2, 3, 4, 5\} \quad (1)$$

where DEM$(\cdot)$ represents the function of DEM, and f_i represents the i^{th} layer in the *DFP*.

1) Difference Enhancement Module: In Siamese networks, a common approach to extract difference features involves computing the absolute difference between feature maps at the same layer [20], [30], [34]. This method emphasizes the disparities between feature maps while disregarding their original information. In order to mitigate this limitation, we introduce the DEM. The structure of DEM is shown in Fig. 6. Here, we demonstrate this process using feature maps $r_i \in \mathbb{R}^{H\times W\times C}$ and $t_i \in \mathbb{R}^{H\times W\times C}$. The DEM initially computes the absolute difference between r_i and t_i to derive simple difference features $s_i \in \mathbb{R}^{H\times W\times C}$. Subsequently, s_i, r_i, and t_i are concatenated along the channel dimension to yield $d_i \in \mathbb{R}^{H\times W\times 3C}$. Finally, to optimize computational efficiency for subsequent network operations, a 1×1 convolution reduces the channel dimension of d_i, resulting in the output $f_i \in \mathbb{R}^{H\times W\times C}$. The described process is formulated as:

$$f_i = \text{Conv}_{1\times1}(\text{Concat}(r_i, t_i, |r_i - t_i|)) \quad (2)$$

where $|\cdot|$ denotes the absolute value, Concat$(\cdot)$ denotes channel-wise concatenation, and Conv$_{1\times1}(\cdot)$ denotes a 1×1 convolution layer.

B. Bidirectional Feature Pyramid Decoder

To address the gradual loss of global information during the decoding process of the single-path decoder, we propose BiD. BiD performs multi-level feature fusion and decoding on the feature pyramid in two directions: the Top-down Pyramid Decode Path (TD-Path) and the Bottom-up Pyramid Decode Path (BU-Path). Illustrated in Fig. 5, TD-Path comprises three layers, whereas BU-Path comprises four layers. Each layer of TD-Path and BU-Path incorporates a Channel-Integrated Feature Fusion Module (CIFFM), whose structure will be detailed in Section IV-B1. The numbers of layers in TD-Path and BU-Path are determined by experiments in Section VI-E1. BiD takes the *DFP* as input, extracting local and global features by processing *DFP* through TD-Path and BU-Path, respectively. Our BiD can be described as follows:

$$L_i = \text{CIFFM}_i^l(f_i, \text{Up}(L_{i+1})), i \in \{1, 2, 3\} \quad (3)$$

$$G_j = \text{CIFFM}_j^g(f_j, \text{MP}(G_{i-1})), j \in \{2, 3, 4, 5\} \quad (4)$$

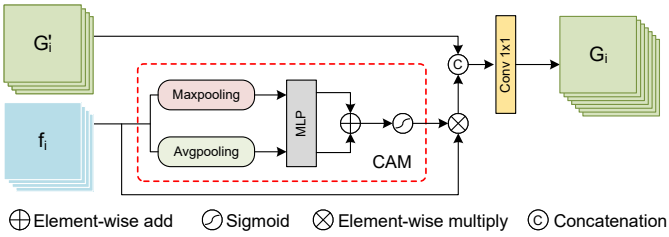


Fig. 7. Structure of CIFFM. CIFFM plays a crucial role in the decode path by integrating different features with the decoded features from the preceding layer. It employs a channel attention module to emphasize the most informative features along this path.

where Up(·) denotes the upsampling function, while MP(·) represents max pooling. CIFFM$_i^l$(·) and CIFFM$_j^g$(·) are the functions of the CIFFM at the i^{th} layer in the TD-Path and the j^{th} layer in the BU-Path, respectively. For clarity and comprehension, i and j correspond to the layer numbers in DFP. For instance, as illustrated in Fig. 5, the first CIFFM in the TD-Path is situated in the third layer of DFP, and thus is labeled CIFFM$_3^l$. If a CIFFM is the first fusion module in a path, it only requires the input of the corresponding layer's feature map from the *DFP*. Finally, G_5 and L_1 are obtained by BiD, respectively. In a feature pyramid generated by a convolutional neural network, lower-level features are typically low-dimensional, high-resolution representations that emphasize local details such as edges and textures, where each pixel corresponds to a small region of the original image. In contrast, higher-level features are high-dimensional and low-resolution, capturing global attributes such as semantic information and object categories, where each pixel represents a larger region of the original image. The TD-Path processes the bottom three feature maps of the DFP, ultimately obtaining the lowest-level features, which primarily capture local information. Conversely, the BU-Path processes the highest-level features, yielding decoded features that primarily encode global information. Therefore, we refer to G_5 and L_1 as the global and local features, respectively.

1) Channel-Integrated Feature Fusion Module: The primary purpose of the CIFFM is to integrate the difference features of the current layer with the decoded features from the previous layer. In BiD, the TD-Path emphasizes local information, contrasting with the BU-Path, which prioritizes global information. Given the varying proportions of local and global information across different channels within difference features, we employ channel attention mechanisms [35] to enhance the fusion of this information. This method enables the network to simultaneously learn both local and global features efficiently.

The structure of CIFFM is shown in Fig. 7. Here, we exemplify the use of CIFFM$_3^g$ at the third layer in BU-Path. CIFFM$_3^g$ accepts inputs $f_3 \in \mathbb{R}^{H\times W\times C}$ and $G_3' \in \mathbb{R}^{H\times W\times C}$, where G_3' is derived from $G_2 \in \mathbb{R}^{2H\times 2W\times C}$ through Max-pooling. CIFFM$_3^g$ directs f_3 into CAM to obtain channel-wise attention weights W_3. Next, CIFFM$_3^g$ multiplies W_3 with f_3 and concatenates the result with G_3' along the channel dimension. Finally, the concatenated features undergo a 1×1 convolution operation to yield $G_3 \in \mathbb{R}^{H\times W\times D}$, where D corresponds to the number of channels for the difference features in the next layer. Our CIFFM can be summarized by the following formulas:

$$W_i = \text{CAM}(f_i) = \sigma(\text{MLP}(\text{MP}(f_i)) + \text{MLP}(\text{AG}(f_i))) \quad (5)$$

$$G_i = \text{Conv}_{1\times 1}(\text{Concat}(G_i', W_i f_i)) \quad (6)$$

where AG(·) represents the average pooling layer. MLP(·) denotes the MLP function, and $\sigma(\cdot)$ denotes the sigmoid function, and CAM(·) denotes the CAM function.

C. Triplet Classifier Prediction Head

Due to the differing information captured by local and global features obtained in BiD, the prediction results of these two features may vary. It is possible for the same pixel location to be predicted as anomalous based on local features, while being predicted as normal based on global features. Therefore, we propose the TCPH, which employs two classifiers to make predictions based on local and global features, respectively, and then utilizes the third classifier to integrate the predictions of the first two classifiers. The structure of TCPH is depicted in Fig. 5.

Here, we assume local features $L \in \mathbb{R}^{256\times 256\times 64}$ and global features $G \in \mathbb{R}^{8\times 8\times 1024}$ as inputs to introduce TCPH. Initially, L is processed by a pixel classifier (clf$_{\text{pi}}$), generating pixel-level predictions $P_{\text{pi}} \in \mathbb{R}^{256\times 256\times 2}$. Concurrently, G is processed by a patch classifier (clf$_{\text{pa}}$), producing patch-level predictions $P_{\text{pa}} \in \mathbb{R}^{8\times 8\times 2}$. The P_{pa} predictions are then upsampled to match the size of P_{pi}, after which the two are concatenated. This concatenated output is then fed into a fusion classifier (clf$_{\text{fu}}$), resulting in the final fusion prediction $P_{\text{fu}} \in \mathbb{R}^{256\times 256\times 2}$. Our TCPH can be described as follows:

$$P_{pi} = \text{clf}_{pi}(L), P_{pa} = \text{clf}_{pa}(G) \quad (7)$$

$$P_{fu} = \text{clf}_{fu}(\text{Concat}(\text{Up}(P_{pa}), P_{pi})) \quad (8)$$

where clf$_{\text{pi}}$ and clf$_{\text{pa}}$ consist of a sequence comprising a 3×3 convolutional layer, ReLU activation function, Batch Normalization, followed by another 3×3 convolutional layer. In contrast, clf$_{\text{fu}}$ is structured with only a 1×1 convolutional layer.

D. Training Procedure

During training, we use cross-entropy loss directly as the loss function for BiSiNet. The total loss is defined as:

$$\begin{aligned} L_{\text{total}} =& L_{\text{ce}}(P_{\text{pi}}, Mask) \\ &+ L_{\text{ce}}(P_{\text{pa}}, \text{Down}(Mask)) \\ &+ L_{\text{ce}}(P_{\text{fu}}, Mask) \end{aligned} \quad (9)$$

where Down(·) downsamples $Mask$ to the size of P_{pa}. The $Mask$ indicates the anomaly mask in the generated sample. The trem L_{ce} represents the cross-entropy loss function, defined as:

$$L_{\text{ce}}(p, y) = -[y\log(p) + (1-y)\log(1-p)] \quad (10)$$

where p is the predicted probability of being anomalous, and y is the ground truth label (0 for normal, 1 for anomaly).

The loss is averaged over all pixels. During each iteration, the model computes the gradients of the loss function with respect to the weights, which are then propagated backward through the network. The optimizer updates the weights to minimize the loss. For further details, please refer to Section VI-C.

The Algorithm 1 outlines the optimization process of the BiSiNet.

Algorithm 1 Pseudo code of BiSiNet

Require: Training dataset X_N with normal image pairs, Cellular Anomaly Generation Strategy $CACS$, Siamese Encoder SE, Bidirectional Feature Pyramid Decoder BiD, pixel classifier clf_{pi}, patch classifier clf_{pa}, fusion classifier clf_{fu}. Set the parameters of learning rate lr, training epochs, and batch size. Dataloader are imported from the Pytorch framework.

Ensure: Trained model.

```
loader = Dataloader(X_N)
for e = 1, 2, ... , epochs do
    for b = 1, 2, ..., len(loader) do
        Step 1: Sample a pair of normal images:
        (I_N^1, I_N^2) = loader[b]
        Step 2: Generate pseudo-samples:
        G_ref, G_tar, Mask = CACS(I_N^1, I_N^2)
        Step 3: Extract Difference Feature Pyramid:
        f_1, f_2, f_3, f_4, f_5 = SE(G_ref, G_tar)
        DFP = {f_1, f_2, f_3, f_4, f_5}
        Step 4: Obtain local features and global features:
        L, G = BiD(DFP)
        Step 5: Pixel-level and patch-level prediction:
        P_pi = clf_pi(L), P_pa = clf_pa(G)
        Step 6: Fusion prediction:
        P_fu = clf_fu(Concat(Up(P_pa), P_pi))
        Step 7: Loss calculation:
        L_total = L_ce(P_pi, Mask) + L_ce(P_pa, Down(Mask)) +
        L_ce(P_fu, Mask)
        Step 8: Optimize network parameters.
    end for
    return Trained model
end for
```

V. Proposed CAGS

Previous methods for generating anomalies often resulted in pseudo anomaly images that deviated markedly from the distribution of real anomaly images [15], [28], [36]. This discrepancy posed a challenge for the network in accurately learning decision boundaries. To mitigate this issue, we propose CAGS. CAGS is designed to generate pseudo images of container markings that closely approximate the distribution of real marking images. Moreover, CAGS is capable of producing four distinct sample types, thereby improving the efficacy of semi-supervised training. In CAGS, we also propose a Pseudo Marking Generation Module (PMGM) designed to synthesize pseudo markings at specified positions within images. Thus, this section first introduce the process of PMGM, followed by a detailed explanation of CAGS.

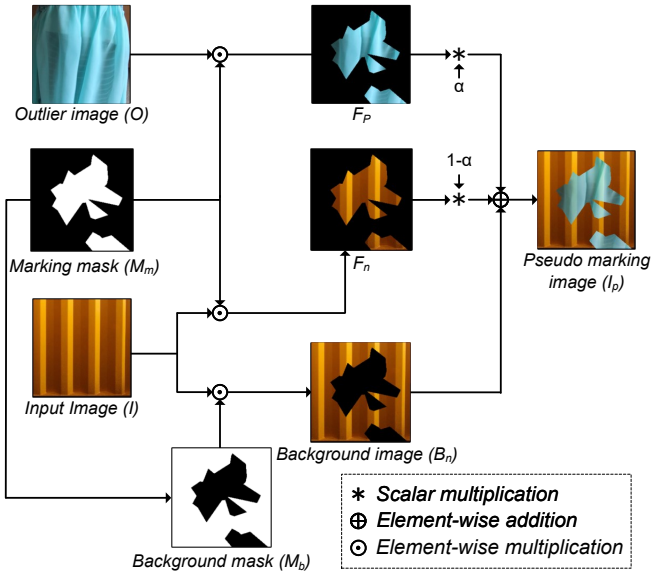


Fig. 8. The process of PMGM. PMGM synthesizes pseudo markings within specified regions indicated by the marking mask in the input image, using patterns sourced from the outlier image.

A. Pseudo Marking Generation Module

Inspired by DRAEM [15], we propose PMGM to address the lack of diversity in collected container marking images. PMGM synthesizes pseudo markings within specified regions indicated by the marking mask in the input image, using patterns sourced from the outlier image. The outlier images were sampled from the Describable Textures Dataset (DTD) [37], which has a different image distribution compared to real container markings. In this study, we adopted a reference-based method, shifting the focus from learning defects in the markings themselves to detecting anomalies by comparing the differences between a reference image and a target image. Consequently, the variations between the outlier images and the real container marking images had minimal impact on the training performance. Furthermore, the DTD comprises various categories with a substantial collection of distinct images. This rich and diverse set enhances the generalization capabilities of semi-supervised training.

As depicted in Fig. 8, PMGM receives an input consisting of an input image I, a marking mask M_m, and an outlier image O. Here, O is from a texture dataset that does not align with I's distribution. M_m specifies the regions where synthetic pseudo container markings are to be applied. To derive the background mask M_b, M_m is inverted. The background image B is obtained by element-wise multiplication of I with M_b. Subsequently, using M_m, separate element-wise multiplications are performed on I and O to obtain F_I and F_o, respectively. Finally, blending B, F_I, and F_o produces the pseudo marking image I_p. Specifically, the pseudo marking image I_p is defined as:

$$I_p = I \odot \text{inverse}(M_m) + (\alpha * \ O + (1-\alpha) * I) \odot M_m \quad (11)$$

where inverse$(\cdot)$ denotes a function that inverts a binary mask. $\odot$ denotes the element-wise multiplication operation

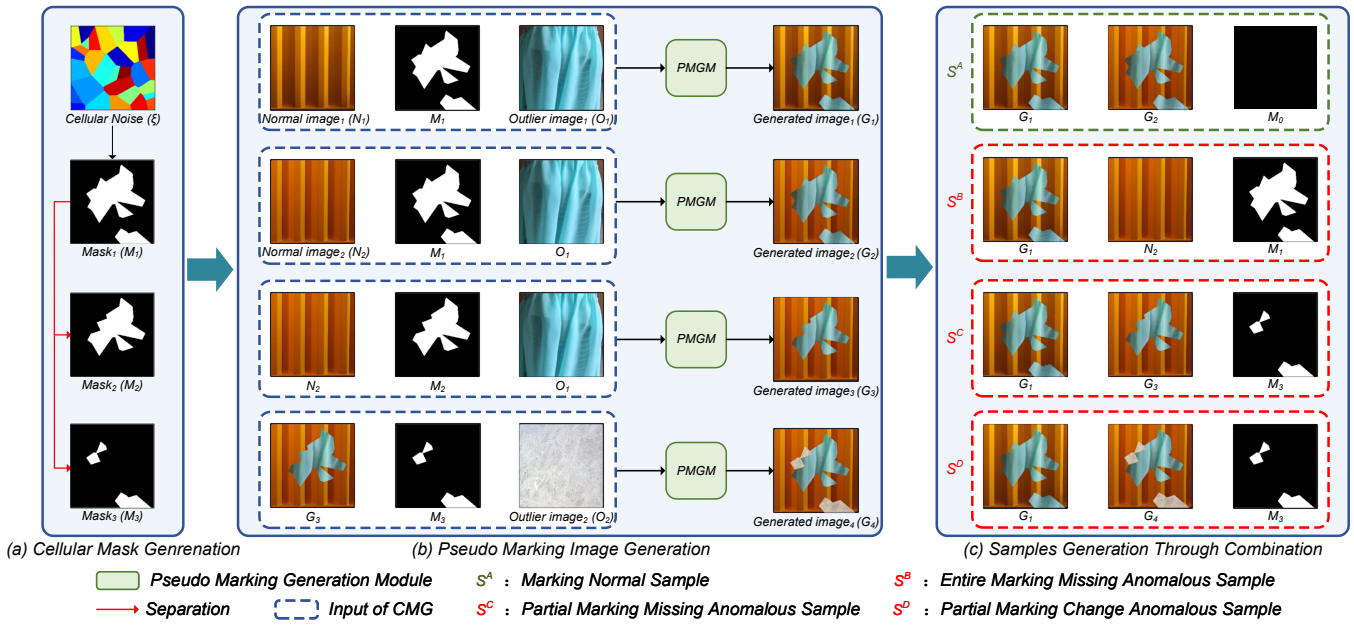


Fig. 9. The three steps of our CAGS. In the first step, three masks are generated based on Cellular Noise. In the second step, PMGM takes normal images, masks, and outlier images from the samples to generate four new pseudo marking images. In the final step, four types of samples are generated by recombining the masks, pseudo marking images, and normal images obtained from the previous steps, including one normal sample and three anomaly samples.

and $*$ indicates the scalar multiplication operation. The opacity parameter, α, is uniformly sampled from the interval $[0.3, 1.0]$.

B. The Process of CAGS

In the previous subsection, we proposed PMGM to generate pseudo container marking images. However, a single image alone cannot serve as a training sample because it is impossible to determine whether the target image is anomalous without a reference image. Therefore, we propose CAGS. CAGS generates training sample triplets, comprising a reference image, a target image, and a pixel-level anomaly mask. As illustrated in Fig. 9, CAGS primarily comprises three steps: Cellular Mask Generation, Pseudo Marking Image Generation, and Sample Generation Through Combination.

1) Cellular Mask Gerenation: In previous methods for generating samples, binarizing Perlin Noise [23] or random matrix regions are commonly used to obtain foreground masks, whereas CAGS opts for Cellular Noise [25]. Perlin Noise relies on a gradient-continuous function, resulting in smooth masks. In contrast, Cellular Noise operates on competitive cell relationships, yielding masks that are more regular and structured. Given that container markings typically exhibit regular and structured features, masks based on Cellular Noise are better suited for the CMAD.

As shown in Fig. 9(a), a two-dimensional Cellular Noise ξ is first generated. ξ comprises N irregular polygonal regions, denoted as $\xi = \{c_i\}_{i=1}^{N}$, where N represents the total number of regions in ξ and c_i represents the i^{th} region in the set. Next, k regions are randomly selected from ξ as foreground regions, denoted as $F = \text{RandomSubset}(\xi, k)$, where $k < N$. The remaining regions constitute the background. Subsequently, Mask M_1 is derived by binarizing based on the foreground and background regions. Therefore, M_1 can be defined as:

$$M_1 = \text{Assign}_1(F) \cup \text{Assign}_0(\xi - F) \tag{12}$$

where the function $\text{Assign}_1(\cdot)$ assigns a value of 1 to the corresponding regions of elements in the input set. Similarly, $\text{Assign}_0(\cdot)$ assigns a value of 0 to the regions represented by elements in the input set.

Finally, a Separation operation is performed on M_1 to derive Mask M_2 and Mask M_3. Specifically, the foreground region set F is randomly partitioned into two subsets, F_1 and F_2, where $F = F_1 \cup F_2$ and $F_1 \cap F_2 = \emptyset$. M_2 and M_3 can be defined as follows:

$$M_2 = \text{Assign}_1(F_1) \cup \text{Assign}_0(\xi - F_1) \tag{13}$$

$$M_3 = \text{Assign}_1(F_2) \cup \text{Assign}_0(\xi - F_2) \tag{14}$$

In the Cellular Mask Generation step, three masks, namely M_1, M_2, and M_3, are obtained for generating the pseudo marking image in the subsequent steps. Their relationship can be expressed as follows, $M_1 = M_2 + M_3$.

2) Pseudo Marking Image Generation: In this step, the process involves three elements: (a) three masks generated in the Cellular Mask Generation step, (b) pairs of aligned sample images for network input, and (c) two outlier images. These components are aggregated and input into PMGM to produce four distinct pseudo marking images. These generated images will then be used in the next step.

In BiSiNet, the input samples during training consist of two aligned normal container images. Fig. 9(b) illustrates normal images N_1 and N_2 as pairs within a single sample. This section does not differentiate between reference and target images within samples, since during training, each image in

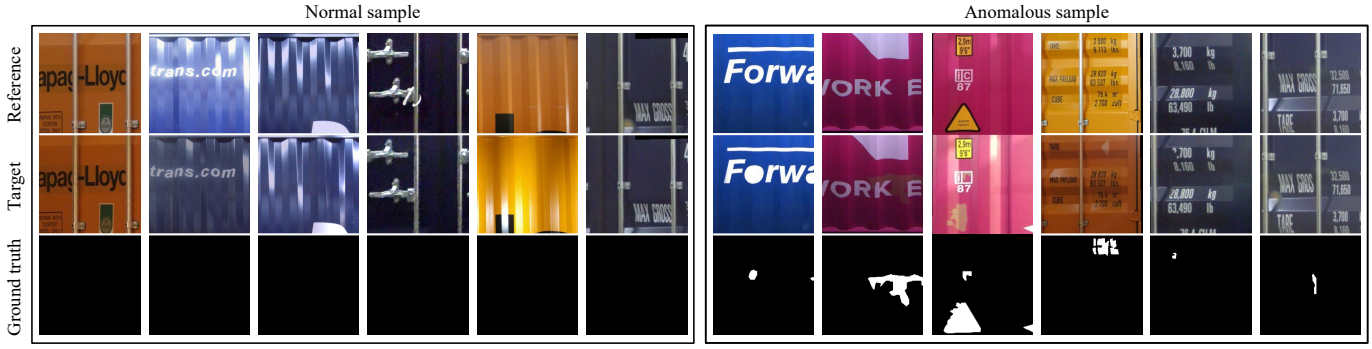


Fig. 10. Examples of normal and anomalous samples from the ContainerMAD dataset.

the sample is a normal image. Outlier images O_1 and O_2 are used to generate various pseudo marking patterns. The four generated pseudo marking images are denoted as G_i, where $i \in \{1, 2, 3, 4\}$. As illustrated in Fig. 9(b), G_i is generated by PMGM. G_i can be formulated as follows:

$$G_1 = \text{PMGM}(N_1, M_1, O_1) \tag{15}$$

$$G_2 = \text{PMGM}(N_2, M_1, O_1) \tag{16}$$

$$G_3 = \text{PMGM}(N_2, M_2, O_1) \tag{17}$$

$$G_4 = \text{PMGM}(G_3, M_3, O_2) \tag{18}$$

where PMGM$(\cdot)$ represent the function of PMGM.

3) Samples Generation Through Combination: In this step, as illustrated in Fig. 9(c), we recombine the images from the previous step with the masks to generate four distinct types of training samples. These samples are categorized into four types: normal marking sample S^A, entire marking missing anomalous sample S^B, partial marking missing anomalous sample S^C, and partial marking change anomalous sample S^D. They are represented by the following formulas: $S^A = \{G_1, G_2, M_0\}$, $S^B = \{G_1, N_2, M_1\}$, $S^C = \{G_1, G_3, M_3\}$, $S^D = \{G_1, G_4, M_3\}$, where M_0 is a fully zero two-dimensional mask.

These diverse generated samples enhance the quality of training by providing a broader range of scenarios and variations. This variety allows the model to learn more effectively, increasing its robustness and generalization capabilities.

VI. Experiment and Result

A. ContainerMAD Dataset

In this study, we developed a dataset named ContainerMAD for detecting anomalies in container markings. Image acquisition was conducted using the CMAD system described in Section III. Images collected from the same production batch and camera were aligned and had their backgrounds removed. The collected data were manually annotated. Each sample in the ContainerMAD dataset comprises a reference image, a target image, and an anomaly mask. Consequently, all images labeled as normal can serve as reference images, while the remaining images can serve as target images. Selecting a normal image as the target image designates the sample as normal, whereas selecting an anomalous image designates the sample as anomalous. We applied a sliding window method for cropping and sampling the data, yielding 145,960 samples, each with a resolution of 256×256 pixels. In ContainerMAD, the training set consists exclusively of normal samples, totaling 138,389. The validation set includes 2,557 samples, with 800 being anomalous. The test set comprises 5,014 samples, including 1,557 anomalous samples, as detailed in Table I. Fig. 10 illustrates samples from ContainerMAD. ContainerMAD presents two primary challenges. First, normal samples contain various interfering factors such as color differences, blurriness, reflections, shadows, and halos, which can be mistaken for anomalies. Second, some anomalous samples have regions that are either lighter in color or smaller in size, making them difficult to detect. To ensure fairness and consistency across experiments, we used the same random seed for all trials and relied on offline data for training. Each experiment was conducted three times, and the reported results reflect the mean values obtained from these repetitions.

TABLE I
Sample Distribution of ContainerMAD Dataset.

Dataset	Anomalous sample	Normal sample	Total
Training set	0	138389	138389
Validation set	800	1757	2557
Test set	1557	3457	5014

B. Evaluation Metrics

We evaluated the ContainerMAD dataset using various metrics: Overall Accuracy (OA), Intersection over Union (IoU), Precision (Pre), Recall (Rec), and F1 Score (F1). These metrics were selected to provide a comprehensive evaluation of the detection system from multiple perspectives. Overall Accuracy measures the model's correctness by calculating the proportion of true results (including true positives and true negatives) out of all cases. Intersection over Union evaluates the overlap between the predicted anomaly region and the ground truth, thereby providing spatial accuracy for the localization of anomalies. Precision quantifies the proportion of true positives among all positive predictions, thereby assessing the relevance of positive results. Recall measures the proportion of actual positives correctly identified by the model, indicating

the model's sensitivity. Finally, the F1 Score, the harmonic mean of Precision and Recall, provides a balanced metric that considers both false positives and false negatives. The definitions of these evaluation metrics are as follows:

$$\mathrm{OA} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}) \tag{19}$$

$$\mathrm{IoU} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP} + \mathrm{FN}) \tag{20}$$

$$\mathrm{Pre} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \tag{21}$$

$$\mathrm{Rec} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{22}$$

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Pre} \cdot \mathrm{Rec}}{\mathrm{Pre} + \mathrm{Rec}} \tag{23}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Evaluations were conducted on container marking anomaly detection (at the image level) and localization (at the pixel level). The detection evaluation measures the overall accuracy of identifying images as either anomalous or normal, while the localization evaluation assesses the accuracy of pixel-level anomaly identification within the images.

C. Implementation Details

All experiments were conducted under identical environmental and hardware configurations. We trained all models on the ContainerMAD dataset for 150 epochs, each with 1000 iterations and a batch size of 8. Applying the SGD optimizer with a momentum of 0.9 and weight decay set at 0.0005. The initial learning rate was 0.001, updated by linear decay. In our experiments, 50% of the training samples underwent processing via CAGS. CAGS generated four types of samples with equal probabilities, and outlier images were sourced from the DTD [37]. In practical industrial applications, the performance of networks in detecting anomalies in container markings at the image level is more crucial than pixel-level anomaly localization. Furthermore, for cost-efficiency reasons, a unified model typically handles both detection and localization tasks in industrial settings. Therefore, unless specified otherwise, we selected the model exhibiting the highest performance in image-level container marking anomaly detection on the validation set and evaluated it on the test set.

D. Comparative experiments

1) Comparisons of the Proposed BiSiNet with Other Networks: Previous anomaly detection methods are generally designed for industrial products with consistent appearances. However, container marking images display considerable variability, rendering these conventional approaches ineffective. Consequently, in our comparative experiments, we examined applicable methods from other relevant fields. U-Net [44] is a structure widely used in the field of semantic segmentation. Therefore, we compared three non-Siamese segmentation networks based on the U-Net architecture with distinct characteristics: FC-EF [30], SwinUnet [38], and UNet++ [43]. In addition, we compared a non-Siamese network that integrates Transformer and CNN architectures, referred to as FCBFormer [40]. Furthermore, reference-based CMAD enables the network to learn the differences between the reference image and the target image, facilitating the identification of normal and abnormal regions. Therefore, we posit that Siamese networks possess an inherent advantage in such tasks. Additionally, the objective of remote sensing change detection is to identify surface changes between two temporal images, which closely aligns with our task. The field of remote sensing change detection represents one of the most advanced applications of Siamese network technology. Consequently, we selected methods from this domain for comparison, including FC-diff [30], FC-conc [30], SNUNet [31], BIT [41], ICIF-Net [42], and LUNet [43]. The characteristics of the various networks are as follows:

- **FC-EF**: A fully convolutional network that employs a U-Net architecture, using concatenated image pairs as input.
- **SwinUNet**: Constructs an encoder-decoder architecture for U-Net using hierarchical Swin Transformer modules.
- **UNet++**: Introduces dense skip connections based on U-Net, thereby enhancing feature reuse and information flow.
- **FCBFormer**: Extracts features through both a transformer branch and a fully convolutional branch, subsequently concatenating the extracted features for prediction.
- **FC-diff**: A fully convolutional network that utilizes Siamese networks to extract features and fuses the features using a subtraction method.
- **FC-conc**: A fully convolutional network that employs Siamese networks to extract features and fuses the features using a concatenation method.
- **SNUNet**: A dense connection Siamese network that integrates the Siamese networks with UNet++.
- **BIT**: Converts pairs of images into a few tokens, models spatial context using a Transformer encoder, and refines the original features through a decoder.
- **ICIF-Net**: Achieves interaction and fusion of local and global features by innovatively integrating CNN and Transformer, thus improving detection accuracy through multi-scale feature integration.
- **LUNet**: Incorporates conventional long short-term memory (Conv-LSTM) into specific convolutional layers of U-Net.

Since CMAD requires input samples to include both a reference and a target image, for non-Siamese networks, we concatenated these images before input. Table II presents the results of different networks on ContainerMAD. In anomaly detection, BiSiNet achieved the highest OA, F1 score, and Precision, reaching 97.13%, 95.3%, and 96.88%, respectively. It outperformed LUNet by 0.6% in OA and 0.95% in F1 score. Although BiSiNet's Recall was 1.35% lower than FC-conc, FC-conc's Precision was only 83.77%, which is 13.11% lower than BiSiNet's. Therefore, BiSiNet is superior in anomaly detection. In anomaly localization, BiSiNet achieved the highest OA, IoU, F1 score, and Recall, surpassing the second-best FC-diff by 0.2%, 4.14%, 2.92%, and 5.04%, respectively. It is noteworthy that LUNet performed well in anomaly detection, with its F1 score being only 0.95% lower than BiSiNet's. However, in anomaly localization, its F1 score was 7.99% lower than BiSiNet's. Conversely, FC-diff excelled in localizing container marking anomalies but showed a significant decline in detecting these anomalies at the image

TABLE II
RESULTS FOR THE TEST SET OF THE CONTAINERMAD DATASET. SELECTING THE MODEL WITH THE HIGHEST PERFORMANCE IN ANOMALY DETECTION ON THE VALIDATION SET. ALL DATA IN THE TABLE IS PRESENTED AS PERCENTAGES (%), WITH THE EXCEPTION OF PARAMS AND FPS.

Method		Detection				Localization					Parm(M)	FPS
		OA	F1	Pre	Rec	OA	IoU	F1	Pre	Rec		
Non-Siamese	FC-EF [30]	95.47	92.67	93.24	92.10	97.87	56.20	71.96	**94.52**	58.10	**1.35**	**320.25**
	SwinUnet [38]	92.02	87.25	86.59	87.93	98.22	63.84	77.93	93.74	66.69	27.15	192.57
	UNet++ [39]	94.97	91.86	92.40	91.33	97.10	40.06	57.21	94.14	41.09	36.63	24.10
	FCBFormer [40]	92.76	87.67	93.01	82.92	98.17	63.23	77.47	92.06	66.88	52.95	41.04
Siamese	FC-diff [30]	94.36	91.18	88.61	93.90	98.32	66.12	79.61	93.31	69.41	**1.35**	253.66
	FC-conc [30]	92.76	89.08	83.77	**95.12**	98.23	63.78	77.88	94.44	66.27	1.55	243.11
	SNUNet [31]	94.30	90.65	92.34	89.02	97.96	58.65	73.93	93.18	61.28	10.20	44.73
	BIT [41]	94.91	92.02	89.70	94.48	97.88	56.64	72.31	93.70	58.88	3.50	101.95
	ICIF-Net [42]	95.41	92.37	95.48	89.47	97.63	51.18	67.71	94.49	52.76	23.84	58.74
	LUNet [43]	96.53	94.35	95.40	93.32	98.01	59.42	74.54	93.85	61.83	8.45	78.89
	BiSiNet(Ours)	**97.13**	**95.30**	**96.88**	93.77	**98.52**	**70.26**	**82.53**	92.57	**74.45**	18.51	208.89

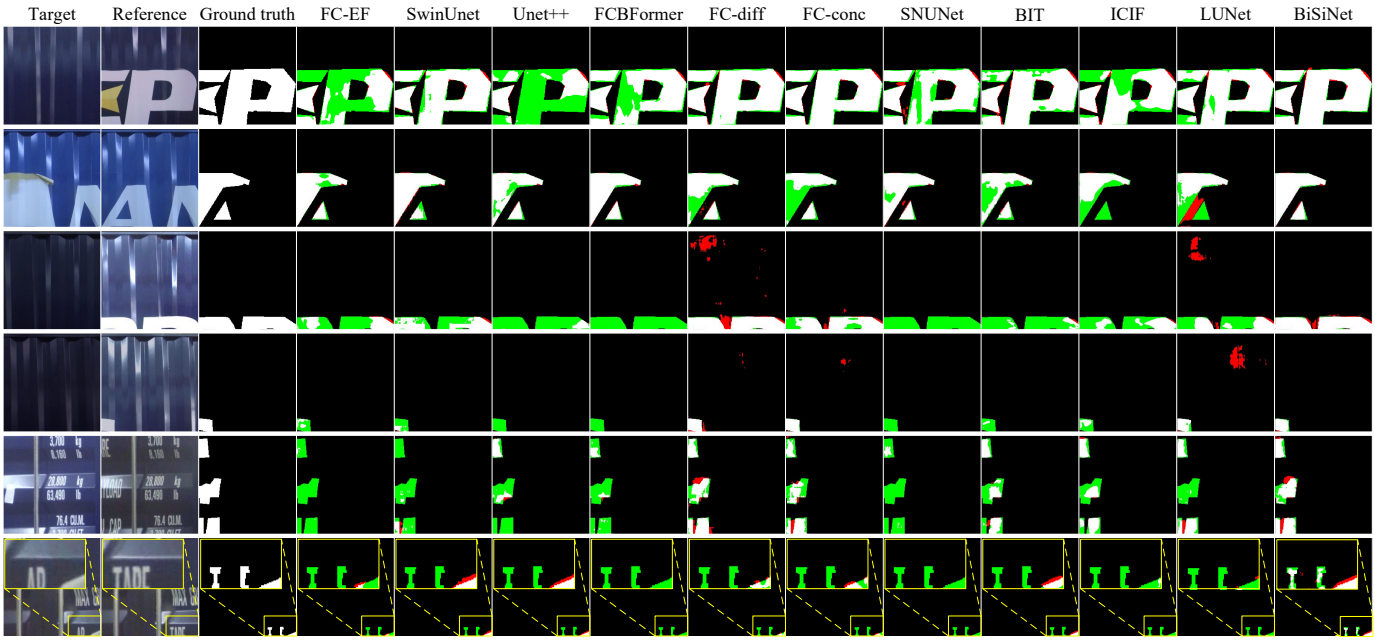


Fig. 11. Visual comparisons between BiSiNet and other methods on the ContainerMAD dataset. In the visualization, white denotes true positives, black denotes true negatives, red denotes false positives, and green denotes false negatives.

level. Overall, BiSiNet achieved the best performance in both container marking anomaly detection and localization.

For qualitative analysis, we visualized the prediction results of different networks and displayed them in Fig. 11. Non-Siamese networks exhibit a notable rate of false negative predictions, indicating their limited effectiveness in reference-based CMAD. Examples from the first and second rows demonstrate BiSiNet's superior performance in localizing larger anomalies. Anomalies positioned at image edges or influenced by reflections, as illustrated in the third and fourth rows, present challenges for both detection and localization. BiSiNet effectively addresses these issues, whereas other networks either miss anomalies or misclassify reflections as anomalies. In the fifth row example, networks like LUNet and SNUnet struggle with reflections and fail to detect the absence of text markings, whereas BiSiNet successfully identifies the anomaly. In the bottom row, where anomalies are small and located at the image edges, presenting a significant challenge, BiSiNet successfully identifies the absence of two small characters, whereas other networks fail on this test case. These visualizations underscore three key strengths of BiSiNet: precise anomaly localization, resilience to interference, and ability to detect anomalies in small regions.

To explore the network's best performance in anomaly localization, we selected the model with the highest localization F1 score on the validation set for evaluation on the test set. The findings are summarized in Table III. The experimental results clearly illustrate that BiSiNet outperforms other networks in both container marking anomaly detection and localization. Consistent with the results presented in Table II, other networks were unable to achieve superior performance in both anomaly detection and localization simultaneously. In contrast, BiSiNet consistently demonstrated the best performance in both areas.

TABLE III
RESULTS FOR THE TEST SET OF THE CONTAINERMAD DATASET. SELECTING THE MODEL WITH THE HIGHEST PERFORMANCE IN ANOMALY LOCALIZATION ON THE VALIDATION SET. ALL DATA IN THE TABLE IS IN PERCENTAGES(%).

Method		Detection		Localization		
		OA	F1	OA	IoU	F1
Non-Siamese	FC-EF [30]	88.97	84.17	98.44	70.26	82.54
	SwinUnet [38]	88.77	83.41	98.38	67.96	80.92
	UNet++ [39]	89.51	84.75	98.46	69.71	82.15
	FCBFormer [40]	91.32	86.36	98.44	69.17	81.78
Siamese	FC-diff [30]	88.57	83.95	98.53	70.98	83.03
	FC-conc [30]	87.58	82.74	98.59	72.32	83.94
	SNUNet [31]	86.86	81.94	98.58	71.70	83.52
	BIT [41]	88.39	83.89	98.56	71.73	83.54
	ICIF-Net [42]	91.01	85.18	97.97	61.22	75.95
	LUNet [43]	93.90	90.67	98.56	71.12	83.12
	BiSiNet(Ours)	**96.51**	**94.31**	**98.64**	**72.62**	**84.14**

To assess the efficiency of the models, we recorded their parameters (Parm) and FPS on the ContainerMAD dataset, as detailed in Table II. While BiSiNet comprises 18.51 million parameters, it achieves an FPS of 208.89, placing fourth among all models. Only a few lightweight models such as FC-EF, FC-diff, and FC-conc exhibit higher FPS. Given BiSiNet's superior performance in container marking anomaly detection, this trade-off is deemed acceptable. Thus, BiSiNet demonstrates a commendable balance between performance and efficiency.

The experimental results indicate that BiSiNet outperforms other methods in both container marking anomaly detection and localization. We summarize the reasons for BiSiNet's superior performance as follows. First, BiSiNet employs a Siamese network for feature extraction, which effectively assesses image similarities and differences, leading to improved performance compared to non-Siamese methods [30], [38]–[40]. Second, BiSiNet utilizes DEM to fuse features from image pairs, preserving both differential and semantic information, which offers advantages over techniques that merely concatenate or subtract features [30], [31]. Third, BiSiNet's bidirectional decoding mechanism captures both local and global features, mitigating the loss of global information commonly associated with unidirectional decoders. This enhancement contributes to superior performance at both the image and pixel levels, surpassing unidirectional methods [30], [31], [38], [39], [43]. Lastly, due to the relatively simple semantics of container marking images, the fully convolutional architecture of BiSiNet streamlines the model, resulting in faster inference times compared to Transformer, LSTM or dense connection methods [31], [38]–[43].

2) Comparisons of the Proposed CAGS with Other Anomaly Generation Strategy: The primary characteristics of anomalies produced by the anomaly generation strategy include the shape and pattern of the anomalies. Currently, the shapes of these anomalies are typically either island-like (Isla.), derived from the binarization of Perlin Noise, or rectangular (Rect.). Anomaly patterns generally originate from outlier (Outl.) images, pure colors, or the original (Orig.) image itself. To demonstrate the effectiveness of CAGS, we compared it with DRAEM [15], MemSeg [22], Cutout [24], and CutPaste [28]. These methods are representative, each exhibiting distinct characteristics as previously mentioned. It is noteworthy that CAGS employs Cellular Noise innovatively to generate masks, resulting in anomalies with irregular polygonal (Poly.) shapes. The characteristics of various anomaly generation strategies are compared in Table IV. The samples generated from various anomaly generation methods are shown in Fig. 12. The results of BiSiNet trained with samples generated using different anomaly generation strategies are presented in Table V. The findings demonstrate that CAGS surpasses alternative methods in performance. All metrics exhibit optimality, with the exception of localization Precision. Although the Precision of the DRAEM and Cutout methods is higher than that of CAGS, their Recall is significantly lower. Thus, the F1 score offers a more equitable basis for comparing these methods. In terms of the localization F1 score, CAGS is 11.33% higher than the second-best method, Cutout. These results indicate that CAGS is more effective and superior to other methods in both container marking anomaly detection and localization.

TABLE IV
THE COMPARISON OF CHARACTERISTICS AMONG VARIOUS ANOMALY GENERATION STRATEGIES.

Method	Anomaly shape			Anomaly patterns		
	Isla.	Rect.	Poly.	Outl.	Orig.	Pure
DRAEM [15]	✓	×	×	✓	×	×
MemSeg [22]	✓	×	×	✓	✓	×
Cutout [24]	×	✓	×	×	×	✓
CutPaste [28]	×	✓	×	×	✓	✓
CAGS(Ours)	×	×	✓	✓	×	×

TABLE V
COMPARISON OF DIFFERENT ANOMALY GENERATION METHODS. ALL DATA IN THE TABLE IS IN PERCENTAGES(%).

Method	Detection				Localization				
	OA	F1	Pre	Rec	OA	IoU	F1	Pre	Rec
DRAEM [15]	94.20	90.68	90.42	90.94	96.70	31.49	47.90	93.71	32.17
MemSeg [22]	92.46	88.14	86.14	90.24	96.74	33.79	50.51	88.46	35.35
Cutout [24]	91.78	85.59	93.94	78.61	97.84	55.28	71.20	**95.52**	56.75
CutPaste [28]	95.29	92.23	94.60	89.98	96.79	35.96	52.89	85.73	38.25
CAGS(Ours)	**97.13**	**95.30**	**96.88**	**93.77**	**98.52**	**70.26**	**82.53**	92.57	**74.45**

E. Ablation study

1) Number of Layers in TD-Path and BU-Path: Each layer in the TD-Path and BU-Path consists of a CIFFM, and the performance of BiSiNet is influenced by the number of layers in both paths. Therefore, we performed an ablation study on the layer counts in both the TD-Path and BU-Path. The experimental results are depicted in Fig. 13. To facilitate comparative analysis of networks with varied structures, we employed the OA metric for anomaly detection and the F1 score metric for anomaly localization. As shown, when the number of CIFFM layers in the TD-Path and BU-Path are set

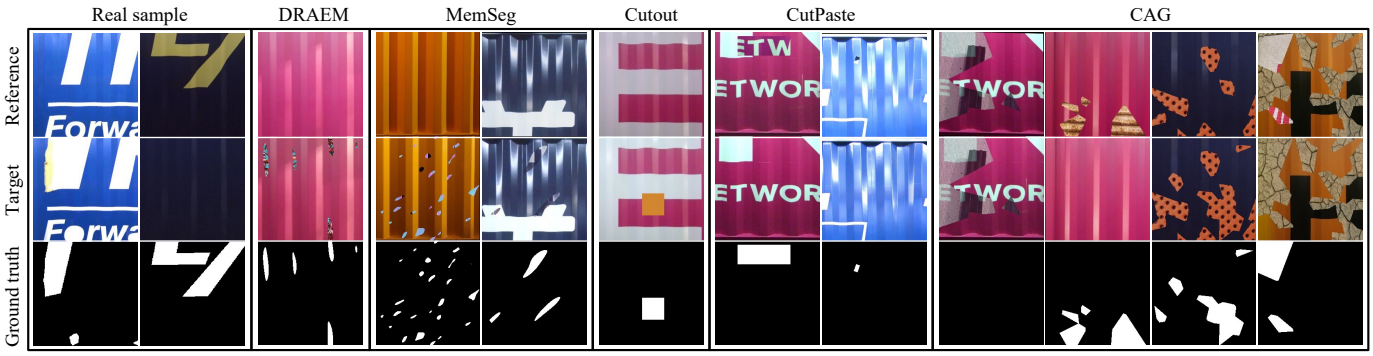


Fig. 12. Visual comparisons of CAGS and other anomaly generation methods on ContainerMAD dataset.

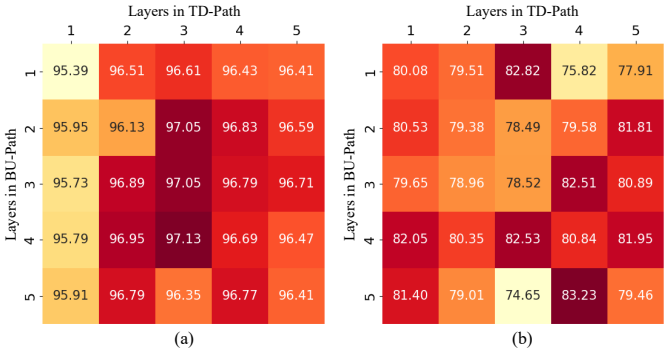


Fig. 13. (a) Detection OA with varying numbers of layers in the BU-Path and TD-Path. (b) Localization F1 with varying numbers of layers in the BU-Path and TD-Path.

to 3 and 4, respectively, BiSiNet achieves top performance in anomaly detection and the second-best performance in anomaly localization. Based on these findings, we configured the TD-Path with 3 layers and the BU-Path with 4 layers.

2) Effectiveness of BiD: To evaluate the effectiveness of BiD's bidirectional dual-path decoding structure, we initially established a baseline method by removing DEM and CIFFM from BiSiNet and eliminating the BU-Path to convert BiD into a single-path decoder. In this configuration, DEM was substituted with a subtraction operation, and CIFFM with a concatenation operation. Since a single-path decoder produces only one local feature map, a pixel classifier was directly employed for prediction in the prediction head. Subsequently, we extended upon the baseline method by incorporating BiD without CIFFM, resulting in a dual-path decoder approach without additional modules. Quantitative comparison results are presented in Table VI(a). It is evident that upon adopting the basic dual-path decoder, the Baseline method exhibited improvement across all evaluation metrics, thereby confirming the efficacy of BiD's bidirectional dual-path decoder structure.

Furthermore, we constructed a UNet-like decoder method similar to UNet [45], replacing BiSiNet's BiD with a decoder using five convolutional layers for decoding and feature fusion via concatenation, denoted as the UNet-like decoder method. Additionally, we removed the BU-path from BiD, converting it into a single-path decoder, to construct another single-path decoder method. The results of the quantitative comparison are

TABLE VI
ABLATION EXPERIMENTS ON THE STRUCTURE AND COMPONENTS OF BISINET. ALL DATA IN THE TABLE IS IN PERCENTAGES(%).

Variants	Detection		Localization		
	OA	F1	OA	IoU	F1
(a) Effectiveness of BiD					
Baseline	94.62	91.65	98.14	61.83	76.42
Baseline + BiD w/o CIFFM	96.77	94.73	98.22	63.47	77.65
UNet-like decoder	95.97	93.56	97.83	55.67	71.52
w/o BU-Path	94.33	91.35	97.85	55.89	71.70
BiD	**97.13**	**95.30**	**98.52**	**70.26**	**82.53**
(b) Effectiveness of DEM					
Subtraction	96.59	94.43	98.16	62.66	77.05
Concatenation	96.39	94.06	98.15	62.94	77.25
DEM	**97.13**	**95.30**	**98.52**	**70.26**	**82.53**
(c) Effectiveness of CIFFM					
Concatenation	96.77	94.66	98.07	60.60	75.47
SAM	96.85	94.84	98.33	66.02	79.54
CBAM	96.95	95.01	**98.55**	**70.73**	**82.86**
CIFFM	**97.13**	**95.30**	98.52	70.26	82.53
(d) Prediction fusion in TCPH					
Pixel-wise AND	96.13	93.59	98.17	62.68	77.06
Pixel-wise OR	96.63	94.48	97.54	60.05	75.04
Fusion classifier	**97.13**	**95.30**	**98.52**	**70.26**	**82.53**
(e) Prediction in TCPH					
Patch-level prediction	96.13	93.59	-	-	-
Pixel-level prediction	96.63	94.48	97.54	60.05	75.04
Fusion prediction	**97.13**	**95.30**	**98.52**	**70.26**	**82.53**

presented in Table VI(a). BiSiNet with BiD outperforms the other two BiSiNets using single-path decoding. In anomaly detection, BiD's OA and F1 scores are higher than those of the single-path decoder. Notably, in anomaly localization, BiD's performance significantly exceeds that of the other two single-path decoder methods. This indicates that BiSiNet can preserve and utilize global information more effectively during decoding, significantly improving the accuracy of anomaly detection and localization.

3) Effectiveness of DEM: The DEM enhances difference features, enabling them to retain more semantic information extracted by the encoder. To validate the effectiveness of DEM, we removed it from BiSiNet and replaced it with subtraction

and concatenation methods separately. The quantitative comparison results are reported in Table VI(b). From these results, in the context of SE within BiSiNet, both subtraction and concatenation methods for feature fusion perform similarly. However, BiSiNet with DEM outperforms these approaches, demonstrating approximately 7% and 5% improvements in localization IoU and F1 score, respectively. This underscores DEM's effectiveness in enhancing the information extracted by SE.

4) Effectiveness of CIFFM: The main function of CIFFM is to enhance the focus of different paths on useful information within the difference features specific to their respective paths. For the TD-Path, CIFFM emphasizes local information within the difference features. Similarly, for the BU-Path, CIFFM emphasizes global information within the difference features. To demonstrate CIFFM's effectiveness, we substituted it with concatenation operation, SAM [35], and CBAM [35], respectively. Table VI(c) presents the results of employing various feature fusion methods in BiD. It is evident that BiSiNet achieves comparable performance with CIFFM and CBAM, with CIFFM performing best in anomaly detection and CBAM showing slightly superior performance in anomaly localization. Given the higher importance of anomaly detection over localization in container production, CIFFM emerges as the preferred choice.

5) Analysis of Prediction Fusion in TCPH: TCPH upsamples the patch-level prediction and concatenates it with the pixel-level prediction, followed by linear fusion using a fusion classifier. Alternatively, performing pixel-wise AND or pixel-wise OR operations directly between the upsampled patch-level prediction and the pixel-level prediction offers a more intuitive fusion method. To analyze the effectiveness of these fusion methods for dual-scale prediction, we compared them with the fusion classifier. The quantitative comparison results are reported in Table VI(d), indicating that using the fusion classifier in TCPH for prediction fusion outperforms both pixel-wise AND and pixel-wise OR operations. Based on these findings, we hypothesize that pixel-wise AND reduces false positives but may increase false negatives, while pixel-wise OR has the opposite effect. Finally the fusion classifier learns the relationship between predictions at these two scales and integrates them to achieve more accurate results.

6) Analysis of Three Different Predictions in TCPH: In TCPH, three classifiers are employed: the patch classifier, the pixel classifier, and the fusion classifier, each generating patch-level, pixel-level, and fusion predictions, respectively. The fusion prediction is derived by integrating the outputs of the other two classifiers. To analyze the relationships among these predictions, we conducted a quantitative comparison, with results presented in Table VI(e). The findings indicate that the fusion prediction, resulting from the integration of predictions at two scales, enhances accuracy in both container marking anomaly detection and localization.

To explore the relationships and roles of different predictions, we visually analyzed anomaly score maps for three predictions, shown in Fig. 14. In normal samples (first and second rows), pixel-level prediction detects anomalies due to alignment and brightness variations, while patch-level prediction yields near-zero scores, unaffected by these factors. Integrating predictions at both scales, fusion prediction effectively addresses these issues. In an anomalous sample (third row), patch-level prediction shows low anomaly scores, contrasting with high scores in pixel-level prediction. Fusion prediction successfully detects the anomaly. These examples demonstrate that, even for more evident anomalies, the patch-level prediction tends to assign lower anomaly scores, whereas the pixel-level prediction assigns higher scores to suspicious regions. Consequently, we conclude that the pixel-level prediction primarily contributes to detecting and locating anomalies, while the patch-level prediction primarily helps eliminate false positives caused by interfering factors.

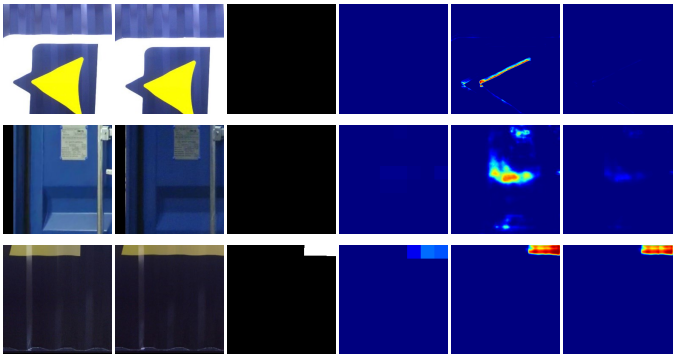


Fig. 14. Anomaly score maps of three different predictions in TCPH. (a) Reference image. (b) Target image. (c) Ground truth. (d) Patch-level prediction. (e) Pixel-level prediction. (f) Fusion prediction.

TABLE VII
ABLATION STUDY OF DIFFERENT TYPES OF SAMPLES GENERATED BY CAGS. ALL DATA IN THE TABLE IS IN PERCENTAGES(%).

Types of generated samples				Detection		Localization		
S^A	S^B	S^C	S^D	OA	F1	OA	IoU	F1
×	✓	×	×	96.59	94.33	98.38	68.67	81.42
✓	✓	×	×	95.93	93.36	98.02	61.38	76.07
×	✓	✓	✓	96.95	94.95	97.72	53.70	69.87
✓	✓	✓	✓	**97.13**	**95.30**	**98.52**	**70.26**	**82.53**

7) Ablation study of CAGS: Our proposed CAGS can generate four types of samples: Marking Normal Sample, Entire marking missing anomalous sample, partial marking missing anomalous sample, and partial marking change anomalous sample, labeled S^A, S^B, S^C, and S^B, respectively. S^C and S^D present challenges due to their mixed normal and anomalous markings. In the ablation experiments, S^C and S^D are grouped together as difficult samples and are not analyzed separately. Table VII presents training results with various sample types. For convenience, CAGS(·) denotes the use of CAGS to generate specified types of samples for training.

The results demonstrate that CAGS(S^A, S^B, S^C, S^D) excels in both container marking anomaly detection and localization. Specifically, compared to CAGS(S^B, S^C, S^D), which has the second-best performance in anomaly detection, CAGS(S^A, S^B, S^C, S^D) improves OA and F1 score by 0.18% and 0.35%, respectively. Although CAGS(S^A, S^B, S^C,

S^D) only slightly improves over the second-best methods at both detection and localization, it is notable that CAGS(S^A, S^B, S^C, S^D) balances performance across both aspects. CAGS(S^B, S^C, S^D) excels in detection performance but falls short in localization, whereas CAGS(S^B) excels in localization but not in detection. CAGS(S^A, S^B) does not perform well in either aspect. This indicates that training with the four types of samples generated by CAGS effectively improves the model's performance in both anomaly detection and localization.

F. Results of BiSiNet on the MVTecAD Dataset

Our proposed BiSiNet is specifically designed for CMAD. To demonstrate BiSiNet's generalization to other image anomaly detection tasks, we modified it and validated its effectiveness on the MVTecAD [46] dataset. The MVTecAD dataset comprises 15 distinct categories of industrial objects, such as carpet, leather, bottles, and screws. Each category contains both normal and anomalous images, with anomalies including scratches, cracks, and stains. This dataset is widely utilized in anomaly detection research within the field of computer vision. We augmented BiSiNet with an encoder-decoder sub-reconstruction network, converting it into a reconstruction-based network, denoted as BiSiNet*. We selected U-Net [44] as the sub-reconstruction network. During the training phase, the input to this sub-reconstruction network consists of synthetic anomaly images, and we employed L2 loss and SSIM loss as reconstruction loss functions. The objective is to enable the model to restore synthetic anomaly images to their original normal forms. After the sub-reconstruction network reconstructs the anomaly images, the resulting images serve as reference images, which are then input to BiSiNet alongside the target images for anomaly detection. BiSiNet* was trained with the same details as DRAEM. The AUROC results for anomaly detection and localization on the MVTecAD dataset are presented in Table VIII. In anomaly detection, BiSiNet* achieved an average AUROC of 98.7%, marking a 0.3% improvement over ATSN [47]. BiSiNet* achieved a 100% AUROC in six categories. However, in anomaly localization, the average AUROC was 96.7%, lower by 0.9% compared to ATSN. This is primarily due to BiSiNet*'s poor anomaly localization performance in the Transistor category, achieving an AUROC of only 83.2%, thus decreasing the overall localization AUROC. Excluding the Transistor category, BiSiNet* and ATSN achieved localization AUROC of 97.7% and 98.0%, respectively, indicating comparable performance. The findings underscore BiSiNet's broad applicability and potential efficacy in detecting and localizing anomalies within industrial contexts.

TABLE VIII
AUROC SCORE (%) RESULTS ON THE MVTECAD DATASET FOR ANOMALY DETECTION/LOCALIZATION. ALL DATA IN THE TABLE IS IN PERCENTAGES(%).

Category	BiSiNet*	DRAEM [15]	CutPaste [28]	ATSN [47]	P-SVDD-C [48]
Carpet	95.7/97.5	97/95.5	93.1/98.3	**99.8/99.1**	94.4/92.9
Grid	**100/99.7**	99.9/**99.7**	99.9/97.5	99.9/99.0	95.6/97.2
Leather	**100/99.7**	**100**/98.6	**100**/99.5	**100**/99.4	96.1/98.2
Tile	**99.9**/96.9	99.6/**99.2**	93.4/90.5	98.8/95.0	93.5/91.9
Wood	**100/97.1**	99.1/96.4	98.6/95.5	99.6/94.8	98/92.1
Average	99.1/**98.2**	99.1/97.9	97/96.3	**99.6**/97.5	95.5/94.5
Bottle	**100**/99	99.2/**99.1**	98.3/97.6	**100**/98.8	99.5/98.6
Cable	96.8/93.2	91.8/94.7	80.6/90	**100/98.0**	97.8/97.6
Capsule	96.6/95.7	**98.5**/94.3	96.2/97.4	96.8/**98.6**	88.7/96.3
Hazlnut	99.9/**98.9**	**100**/99.7	97.3/97.3	**100/98.9**	97.9/98.2
Metal nut	**100**/98	98.7/**99.5**	99.3/93.1	98.7/96.0	96.5/98.1
Pill	98.1/**98.5**	**98.9**/97.6	92.4/95.7	96.7/98.0	91.9/92.4
Screw	96.4/96.8	93.9/97.6	86.3/96.7	**96.8/99.1**	83.3/95.3
Toothbrush	99.2/**99.3**	**100**/98.1	98.3/98.1	93.1/98.9	95.6/96
Transistor	97.3/83.2	93.1/90.9	95.5/93	**98.3/94.5**	92.1/93.5
Zipper	**100**/96.9	**100**/98.8	99.4/**99.3**	98.2/98.3	95.9/96
Average	**98.4**/95.9	97.4/97	94.4/95.8	97.9/**97.9**	93.9/96.2
Average	**98.7**/96.7	98/97.3	95.2/96	98.4/**97.8**	94.5/95.6

VII. CONCLUSION

In this study, we propose an innovative CMAD framework, BiSiNet. BiSiNet utilizes SE to extract difference features between reference images of standard container markings and target images. It then employs BiD for bidirectional dual-path decoding to obtain both local and global features, addressing the issue of progressive global information loss that occurs with traditional single-path decoders. Subsequently, TCPH performs pixel-level and patch-level predictions with two classifiers, while a third classifier integrates these multi-scale predictions to resolve inconsistencies between different scale prediction results. Thanks to BiSiNet's ability to handle both local and global information, it not only possesses strong anti-interference capabilities but also demonstrates accurate anomaly localization. Additionally, we introduce an innovative CAGS to address the lack of diversity in CMAD data and the difficulty of acquiring anomalous data. CAGS uses Cellular Noise to create marking masks, producing pseudo marking images that closely resemble real container marking images, thereby improving the network's ability to learn accurate decision boundaries. CAGS can generate four types of samples—one normal and three anomalous—enhancing network training. To evaluate our method, we developed the ContainerMAD dataset for CMAD. Comparative results indicate that BiSiNet and CAGS outperform other methods in detecting and localizing container marking anomalies. The bidirectional dual-path decoding enables BiSiNet to excel in both tasks within a single model. BiSiNet consistently demonstrates strong performance in anomaly detection and localization while maintaining high efficiency, making it suitable for practical production applications. Furthermore, BiSiNet has shown effective performance on the MvtecAD dataset, confirming its ability to detect anomalies across various industrial products.

In the future, we aim to apply the BiSiNet framework for anomaly detection in other industrial products. A key challenge is the absence of reference images for many products, making BiSiNet unsuitable for these applications. To address this issue, we propose two potential solutions: first, integrating a reconstruction network into BiSiNet to use reconstructed images as references, as discussed in Section VI-F; second, utilizing image retrieval methods to find appropriate images from the database as references for the target image.

REFERENCES

[1] R. Zhang, Z. Bahrami, T. Wang, and Z. Liu, "An adaptive deep learning framework for shipping container code localization and recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2020.

[2] Z. Bahrami, R. Zhang, R. Rayhana, and Z. Liu, "An hrcr-cnn framework for automated security seal detection on the shipping container," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[3] Y. Xu, Z. Liang, Y. Liang, X. Li, W. Pan, J. You, Z. Long, Y. Zhai, A. Genovese, V. Piuri *et al.*, "Data-driven container marking detection and recognition system with an open large-scale scene text dataset," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

[4] S. Jakovlev, T. Eglynas, and M. Voznak, "Application of neural network predictive control methods to solve the shipping container sway control problem in quay cranes," *IEEE Access*, vol. 9, pp. 78 253–78 265, 2021.

[5] Z. Wang, J. Gao, Q. Zeng, and Y. Sun, "Multitype damage detection of container using cnn based on transfer learning," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 5395494, 2021.

[6] Z. Bahrami, R. Zhang, R. Rayhana, T. Wang, and Z. Liu, "Optimized deep neural network architectures with anchor box optimization for shipping container corrosion inspection," in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2020, pp. 1328–1333.

[7] Z. Bahrami, R. Zhang, T. Wang, and Z. Liu, "An end-to-end framework for shipping container corrosion defect inspection," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[8] Z. E. İmamoğlu, T. Tuğlular, and Y. Baştanlar, "Container damage detection and classification using container images," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 2020, pp. 1–4.

[9] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 474–11 481.

[10] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[11] M. Huang, Y. Liu, Z. Peng, C. Liu, D. Lin, S. Zhu, N. Yuan, K. Ding, and L. Jin, "Swintextspotter: Scene text spotting via better synergy between text detection and text recognition," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4593–4603.

[12] A. Y. X. Tham and C. S. Chin, "Yolov5 and residual network for intelligent text recognition on degraded serial number plates," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2024, pp. 301–314.

[13] J. Wang, L. Zhu, X. Yi, and T. Lu, "End-to-end recognition algorithm for degraded invoice text," in *2023 16th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2023, pp. 1–6.

[14] D.-M. Tsai and P.-H. Jen, "Autoencoder-based anomaly detection for surface defect inspection," *Advanced Engineering Informatics*, vol. 48, p. 101272, 2021.

[15] V. Zavrtanik, M. Kristan, and D. Skočaj, "Draem-a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 8330–8339.

[16] H. Guo, L. Ren, J. Fu, Y. Wang, Z. Zhang, C. Lan, H. Wang, and X. Hou, "Template-guided hierarchical feature restoration for anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6447–6458.

[17] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," *arXiv preprint arXiv:2005.02357*, 2020.

[18] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 318–14 328.

[19] J. Hyun, S. Kim, G. Jeon, S. H. Kim, K. Bae, and B. J. Kang, "Reconpatch: Contrastive patch representation learning for industrial anomaly detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 2052–2061.

[20] W. Zhang, Y. Hu, H. Shan, and E. Liu, "An online automatic carbide insert high-resolution surface defect detection system based on template-guided model," *Expert Systems with Applications*, vol. 238, p. 122089, 2024.

[21] H. Wang, J. Zhang, Y. Tian, H. Chen, H. Sun, and K. Liu, "A simple guidance template-based defect detection method for strip steel surfaces," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2798–2809, 2018.

[22] M. Yang, P. Wu, and H. Feng, "Memseg: A semi-supervised method for image surface defect detection using differences and commonalities," *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105835, 2023.

[23] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[24] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[25] S. Worley, "A cellular texture basis function," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 291–294.

[26] X. Li, J. Jing, J. Bao, P. Lu, Y. Xie, and Y. An, "Otb-aae: Semi-supervised anomaly detection on industrial images based on adversarial autoencoder with output-turn-back structure," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–14, 2023.

[27] R. Schirrmeister, Y. Zhou, T. Ball, and D. Zhang, "Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 038–21 049, 2020.

[28] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "Cutpaste: Self-supervised learning for anomaly detection and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9664–9674.

[29] C. Chen, Q. Wu, J. Zhang, H. Xia, P. Lin, Y. Wang, M. Tian, and R. Song, "U2d2pcb: Uncertainty-aware unsupervised defect detection on pcb images using reconstructive and discriminative models," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–10, 2024.

[30] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 2018, pp. 4063–4067.

[31] S. Fang, K. Li, J. Shao, and Z. Li, "Snunet-cd: A densely connected siamese network for change detection of vhr images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] X. Tang, T. Zhang, J. Ma, X. Zhang, F. Liu, and L. Jiao, "Wnet: W-shaped hierarchical network for remote sensing image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.

[34] Y. Zheng and L. Cui, "Defect detection on new samples with siamese defect-aware attention network," *Applied Intelligence*, vol. 53, no. 4, pp. 4563–4578, 2023.

[35] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[36] T. D. Tien, A. T. Nguyen, N. H. Tran, T. D. Huy, S. Duong, C. D. T. Nguyen, and S. Q. Truong, "Revisiting reverse distillation for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 24 511–24 520.

[37] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3606–3613.

[38] 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 *European conference on computer vision*. Springer, 2022, pp. 205–218.

[39] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer, 2018, pp. 3–11.

[40] E. Sanderson and B. J. Matuszewski, "Fcn-transformer feature fusion for polyp segmentation," in *Annual conference on medical image understanding and analysis*. Springer, 2022, pp. 892–907.

[41] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2021.
[42] Y. Feng, H. Xu, J. Jiang, H. Liu, and J. Zheng, "Icif-net: Intra-scale cross-interaction and inter-scale feature fusion network for bitemporal remote sensing images change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.
[43] M. Papadomanolaki, M. Vakalopoulou, and K. Karantzalos, "A deep multitask learning framework coupling semantic segmentation and fully convolutional lstm networks for urban change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 9, pp. 7651–7668, 2021.
[44] 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.
[45] ——, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
[46] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1038–1059, 2021.
[47] J. Zhu, P. Yan, J. Jiang, Y. Cui, and X. Xu, "Asymmetric teacher–student feature pyramid matching for industrial anomaly detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–13, 2024.
[48] J.-Y. Ahn and G. Kim, "Application of optimal clustering and metric learning to patch-based anomaly detection," *Pattern Recognition Letters*, vol. 154, pp. 110–115, 2022.

Yikui Zhai (Senior Member, IEEE) received his Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in June 2013.

Since October 2007, he has been working with Wuyi University, Jiangmen, China, where he is a Full Professor now. He is also Associate Dean of the School of Electronics and Information Engineering in Wuyi University, since 2021. He has been a Visiting Scholar with Department of Computer Science, the Universita degli Studi di Milano, Italy, during June 2016 to June 2017, August 2023 and January 2024.

His research interests include: Image Processing, Deep Learning, Optical Character Recognition, Object Detection, UAV Change Detection, Self-Supervise Learning.

Wenfeng Pan (Graduate Student Member, IEEE) received the B.S. degree from Wuyi University, Jiangmen, China, in 2019. He is pursuing themaster's degree with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, China.

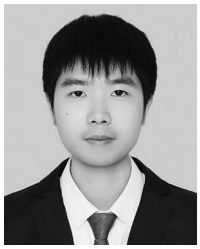
His research interests include anomaly detection, change detection and pattern recognition.

Yanyang Liang received the B.S. degree in automation from the Southwest University of Science and Technology, Mianyang, China, in 2002, and the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2008.,He is currently an Associate Professor with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, Guangdong, China.

His current research interests include computer vision, motion controls, and robust control for nonlinear systems.

Hufei Zhu received the B.E. in electrical engineering from the University of Science and Technology of China in 1998, the M.E. in computing from the National University of Singapore in 2004, and the Ph.D. in electrical engineering from Shanghai Jiao Tong University in 2014. He is currently a Research Fellow at the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, and a Distinguished Professor at Wuyi University, China.

His research interests include neural networks, signal processing, and wireless communications.



Zhihao Long received the B.S. degree from Wuyi University, Jiangmen, China, in 2022. He is currently working toward the master's degree with the School of Electronics and Information Engineering, Wuyi University.

His research interests include computer vision, image generation, and image processing.

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

From 2019 to 2022, he was a post-doctoral researcher at the Università degli Studi di Padova, Italy. He is currently an Assistant Professor at the Department of Computer Science of the Università degli Studi di Milano, Italy. He is Co-chair of the Intelligent Measurement Systems Technical Committee (TC-22) of the IEEE Instrumentation and Measurement Society since 2023.

His research activities are focused on theoretical, methodological, and applied aspects of computational intelligence for signal and image processing.

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

He has been a Visiting Researcher with the University of Toronto, Toronto, ON, Canada. Original results have been published in over 30 papers in international journals, proceedings of international conferences, books, and book chapters. His current research interests include signal and image processing, three-dimensional reconstruction, computational intelligence technologies for biometric systems, industrial and environmental monitoring systems, and design methodologies and algorithms for self-adapting systems.

Vincenzo Piuri (Fellow, IEEE) received the M.S. and Ph.D. degrees in computer engineering from Politecnico di Milan, Milan, Italy, in 1984 and 1988, respectively.

He was the Department Chair with the University of Milan, Milan, from 2007 to 2012, where he has been a Full Professor since 2000. He was an Associate Professor with Politecnico di Milan from 1992 to 2000, a Visiting Professor with The University of Texas at Austin, Austin, TX, USA from 1996 to 1999, and a Visiting Researcher with George Mason University, Fairfax, VA, USA, from 2012 to 2016. He founded a startup company, Sensure srl, Bergamo, Italy, in the area of intelligent systems for industrial applications (leading it from 2007 to 2010) and was active in industrial research projects with several companies. His main research and industrial application interests are intelligent systems, computational intelligence, pattern analysis and recognition, machine learning, signal and image processing, biometrics, intelligent measurement systems, industrial applications, distributed processing systems, Internet-of-Things, cloud computing, fault tolerance, application-specifi digital processing architectures, and arithmetic architectures.

Dr. Piuri is an ACM Fellow.

Fabio Scotti (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the Politecnico di Milano, Milan, Italy, in 2003.

He was an Assistant Professor with the Department of Information Technologies, Università degli Studi di Milan, Milan, from 2002 to 2015, where he was an Associate Professor with the Department of Computer Science from 2015 to 2020. He has been a Full Professor with the Università degli Studi di Milan since 2020. Original results have been published in over 150 papers in international journals, proceedings of international conferences, books, book chapters, and patents. His current research interests include biometric systems, machine learning and computational intelligence, signal and image processing, theory and applications of neural networks, 3-D reconstruction, industrial applications, intelligent measurement systems, and high-level system design.

Prof. Scotti is an Associate Editor of the IEEE Transactions on Human–Machine Systems and the IEEE Open Journal of Signal Processing. He is serving as a Book Editor (Area Editor, section Less-Constrained Biometrics) of the Encyclopedia of Cryptography, Security, and Privacy (3rd Edition, Springer). He has been an Associate Editor of the IEEE Transactions on Information Forensics and Security, Soft Computing (Springer) and a Guest Coeditor of the IEEE Transactions on Instrumentation and Measurement.