Multi-Scale CNN-Transformer Hybrid Network for Rail Fastener Defect Detection

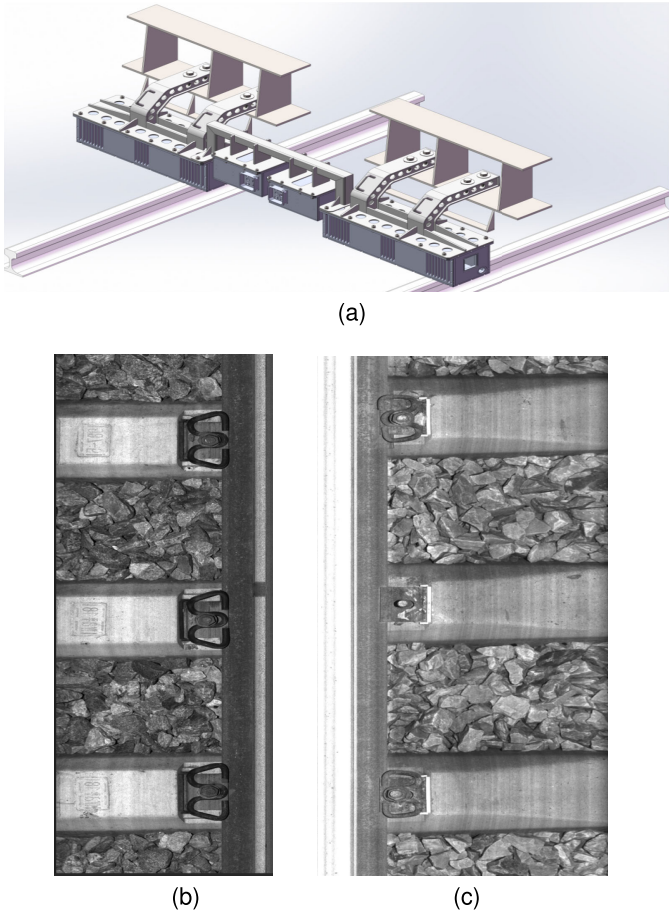
Jin He, Wei Wang, Fengmao Lv, *Member, IEEE*, Haonan Luo, Gexiang Zhang, *Senior Member, IEEE*, and Zhenghua Chen, *Senior Member, IEEE*

***Abstract*— Defect detection in rail fasteners is crucial for train safety, as defective fasteners can cause derailments and severe safety incidents. However, Existing algorithms often struggle in various real-world scenarios due to challenges such as obscured fasteners, motion blur in images, varying camera angles, and fasteners submerged in water. To address these challenges, we propose a Multi-scale CNN-Transformer Hybrid Network for Rail Fastener Defect Detection (MCHNet-RF2D), specifically designed to identify fastener defects in complex environments. Our approach constructs an efficient CNN block and a multi-scale Vision Transformer block to alternately extract local detail features and global semantic features of the fasteners. These features are seamlessly integrated through multi-scale fusion to enhance defect recognition robustness. By combining comprehensive global recognition with detailed local defect detection, MCHNet-RF2D outperforms existing CNN-Transformer hybrid networks by 2.8% and surpasses current fastener defect detection algorithms by 2.9%. In practical deployment on over 40 trains, our model successfully detected more than 2,000 fastener defects, demonstrating its effectiveness in diverse and challenging conditions.**

***Index Terms*— Defect detection, rail fastener, CNN-Transformer, hybrid network, multi-scale fusion, real-world scenarios.**

I. Introduction

RAIL fasteners play a vital role in securing the steel rail to the track bed, preventing lateral and longitudinal displacement of the rail and thus fostering a safe and dependable operating environment for trains. China's extensive railway network relies heavily on these fasteners to keep the steel rails in place. However, continuous vibrations and jolts, coupled with long-term exposure to harsh weather and natural environments, can weaken the intended strength and stability of the rail fastener structure, eventually leading to issues such as loss of clips, breakage, and missing spikes. Any anomalies in the fasteners can result in serious safety accidents, such as train derailments, due to the inability to properly secure the rails. Therefore, rail fastener defect detection has become critical for railway safety operations.



(a) (b) (c)

Fig. 1. a) Rail inspection equipment mounted on the train. b) Image of the left side of the track. c) Image of the right side of the track.

A. Rail Inspection Equipment and Multi-Type Fastener Defects

Rail inspection equipment (Fig. 1(a)) is installed on the underside of the train. As the train moves at high speed, this equipment captures rail images, as depicted in Fig. 1(b)(c). There are numerous types of rail fasteners, primarily classified into two categories based on the clip type on the fastener: E-type clip fasteners (Fig. 2(a)) and W-type clip fasteners (Fig. 2(b)). The main defects associated with E-type clip

Received 21 June 2024; revised 1 December 2024; accepted 7 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62106204, in part by the Fundamental Research Funds for the Central Universities under Grant 2682024ZTPY055, in part by the Key Research and Development Program of Sichuan Province under Grant 2023YFG0180, in part by the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0462, Grant 2023NSFSC1985, and Grant 2025YFHZ0124, in part by the Transfer and Transformation Demonstration Project of Scientific and Technological Achievements of Sichuan Province of China under Grant 2024ZHG0017, in part by the Research Foundation of Chengdu University of Information Technology under Grant KYTZ2023018, and in part by the Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education. The Associate Editor for this article was Q. He. *(Corresponding author: Wei Wang.)*

Jin He, Wei Wang, and Gexiang Zhang are with the School of Automation, Chengdu University of Information Technology, Chengdu 610225, China (e-mail: hejin1977@cuit.edu.cn; wangwei83@cuit.edu.cn; zhgxdylan@126.com).

Fengmao Lv and Haonan Luo are with the School of Computer and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China (e-mail: fengmaolv@126.com; lhn@swjtu.edu.cn).

Zhenghua Chen is with the Institute for Infocomm Research (I^2R), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: chen0832@e.ntu.edu.sg).

Digital Object Identifier 10.1109/TITS.2025.3540846

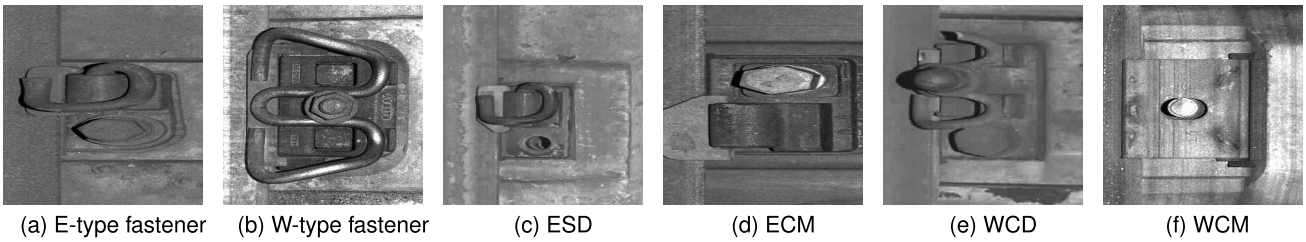


Fig. 2. Depicts various types of fasteners along with their corresponding defects. (a) represents the E-type clip fastener; (b) represents the W-type clip fastener; (c) and (d) respectively indicate spike damage and clip missing in the E-type fastener; (e) and (f) denote clip damage and clip missing in the W-type fastener.

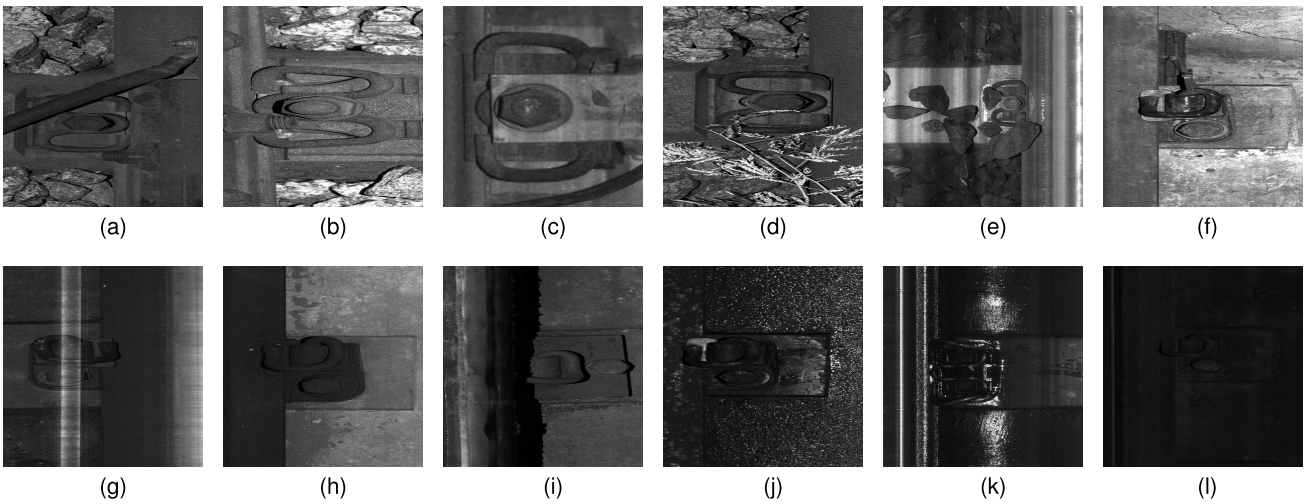


Fig. 3. Depicts the challenges of fastener defect detection in various complex real-world scenes. (a)-(f) illustrate fasteners obstructed by various objects. (g) shows motion blur caused by high-speed movement. (h) depicts rust on fasteners. (i) shows partial visibility due to camera angles. (j) displays blurred images due to water submersion. (k) and (l) demonstrate image blurriness caused by oil stains on fasteners and inadequate camera illumination, respectively.

fasteners include Spike Drop (ESD) (Fig. 2(c)) and Clip Missing (ECM) (Fig. 2(d)). For W-type clip fasteners, the common defects are Clip Damage (WCD)(Fig. 2(e)) and Clip Missing (WCM) (Fig. 2(f)).

B. The Challenge of Identifying Fastener Defects in Complex Real-World Scenes

The extensive mileage and broad coverage of railway and subway networks in China pose diverse challenges for fastener defect detection. Fig. 3 showcases some of these scenarios. Objects like cables, track clamps, and stones can obstruct fasteners, as shown in Fig. 3(a)-(f). The high-speed movement of trains causes motion blur and trailing images (Fig. 3(g)), while rust on fasteners leads to darkened images that are difficult to identify (Fig. 3(h)). Additionally, fasteners can be obscured due to camera angle (Fig. 3(i)), and blurriness can be caused by water on fasteners (Fig. 3(j)), oil stains (Fig. 3(k)) and inadequate camera illumination (Fig. 3(l)). These challenges significantly increase the difficulty of detecting fastener defects on railway rails.

C. Current Rail Fastener Detection Algorithms and Their Issues

Recently, machine vision-based algorithms for fastener defect detection have gained significant attention [1], [2], [3]. For example, Liu et al. [4] employed a cycle-consistent adversarial network to generate W-type defective fasteners and validated their approach by combining generated and real defective fastener images for model training and evaluation, demonstrating promising results. Similarly, Qiu et al. [5] applied style transfer techniques to synthesize defective fastener samples and utilized a modified YOLOv8-fam model for defect detection. Su et al. [6] tackled the challenge of sample imbalance by leveraging geometric constraints and image inpainting to generate defective fastener samples, thereby enhancing defect detection accuracy. However, despite their notable contributions, these methods focus on generating a single type of defective fastener (W-type) and fail to account for defect generation in complex scenarios, which limits their applicability in identifying fastener defects under challenging real-world conditions. Xiao et al. [7] proposed an improved Faster R-CNN network to address the issue of three critical missing small fasteners. Liu et al. [8] applied a modified Cascade Faster R-CNN to tackle fastener defect detection. However, inadequate feature extraction within the backbone network has constrained the performance improvements of these models in complex real-world scenarios.

Moreover, these algorithms rely heavily on CNNs and their variants, which inherently lack the ability to establish long-range dependencies and achieve a global perception understanding [9], [10]. Consequently, they exhibit low recall and precision in recognizing fastener defects across various complex scenarios. Moreover, fastener defect detection faces additional challenges, such as recognizing multiple defect types and achieving robust performance in

intricate natural environments. Consequently, existing algorithms are increasingly inadequate in addressing these issues effectively.

D. Challenges in Applying Generic Defect Detection to Fastener Defect Detection

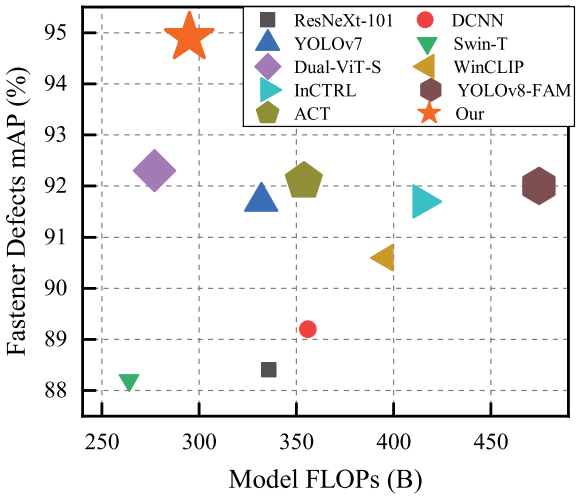
With the recent advancements in ViT Transformers [11], [12] and text-image multimodal [13] development, there has been a surge in Generic Defect Detection (GDD) algorithms. For instance, WinCLIP [14] enhanced the CLIP network by incorporating a compositional ensemble of state words and prompt templates, aligning defect image features with prompt text for zero/few-shot learning defect detection. Zhu and Pang [15] introduced an in-context residual learning model for generalist anomaly detection, achieving excellent performance on open-source defect datasets using few-shot sample prompts. Additionally, there are other generic zero/few-shot defect detection algorithms [16], [17], [18] based on single-class or multi-class classification.

However, directly applying these GDD algorithms to fastener defect detection results in low precision and recall. The core issue is that fastener images are derived from complex real-world natural environments, while open-source defect datasets [19], [20] used by the GDD algorithm, despite being from industrial domains, have relatively clean and uniform backgrounds. Consequently, due to substantial external interference, GDD algorithms are inadequate for defect detection in complex natural conditions.

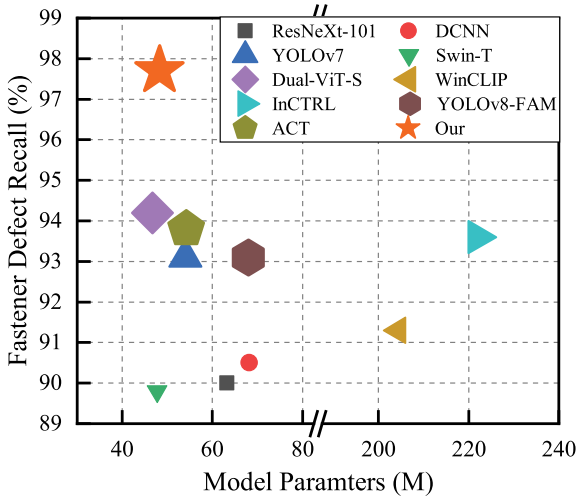
Existing fastener detection algorithms and GDD algorithms fail to adequately detect fastener defects in complex real-world natural environments. To address this issue, we propose MCHNet-RF2D, an object detection-based network that leverages Transformers' proficiency in extracting global semantic features (i.e., target objects) and CNNs' expertise in capturing local detailed features (i.e., object details). MCHNet-RF2D alternately extracts local and global features, and fuses them at multiple scales. The fused features possess both the ability to recognize the global semantics of fasteners and to represent the local details of fastener defects. Compared to GDD and existing fastener detection algorithms, MCHNet-RF2D demonstrates significantly improved precision and recall in defect recognition, as shown in Fig. 4, while maintaining lower computational overhead and weight parameters.

The main contributions of this paper are:

- Proposing a novel multi-scale CNN-Transformer hybrid network for the first time to detect fastener defects.
- Overcoming the challenges of low recall and precision in detecting rail fastener defects under complex real-world conditions.
- Demonstrating the versatility of MCHNet-RF2D in detecting various railway defects, thereby offering valuable insights for tackling defect detection in complex real-world environments.



(a) FLOPs vs. fastener defect mAP



(b) Weight parameters vs. fastener defect recall

Fig. 4. Compared to state-of-the-art networks, MCHNet-RF2D outperform them while having fewer parameters and FLOPs.

II. Related Work

A. Multi-Scale Network

In object detection, a multi-scale network is used either in the backbone or the neck.

1) Backbone: EfficientViT [21] employed cascaded group attention and introduced a cross-group interaction mechanism to enable efficient fusion of multi-scale information. MViT [22] incorporated multi-scale feature pyramids into the Transformer architecture, resulting in the multi-scale vision transformers model, which offers a more efficient solution for vision transformers. CrossViT [23] adopted a dual-branch transformer structure and integrated multi-scale features through a cross-scale attention mechanism, specifically designed for image classification. MPViT [24] addresses the challenge of modeling multi-scale information in dense prediction tasks by employing a multi-path structure and a cross-path interaction mechanism.

MCHNet-RF2D exhibits multi-scale characteristics in two key aspects: 1) It proposes Efficient Multi-headed

Self-attention (E-MHSA) and Mixture Of Experts (MOE) to extract multi-scale global semantic features, enabling the recognition of fasteners at various scales in complex real-world scenes. 2) It uses a multi-scale fusion of global and local features to create complementary mixed features, thereby improving the precision and recall of fastener defect recognition.

2) Neck: The Feature Pyramid Network (FPN) [25] adopts a top-down approach for multi-scale feature fusion, while PaNet [26] extends FPN by adding a bottom-up layer to achieve bidirectional path fusion. Building on PaNet, biFPN [27] treats the bidirectional paths as the neck layer and stacks this layer repeatedly to ensure feature fusion.

The neck component of MCHNet-RF2D builds upon the biFPN framework. It first performs a bidirectional fusion of global features in the neck layer, followed by integrating scale-specific local features into the global features at each scale. This design achieves comprehensive global feature fusion while effectively integrating local and global features across scales.

B. CNN-Transformer Fusion Algorithm

CNN excels at extracting local features, while Transformer excels at extracting global features. Recently, numerous algorithms [28], [29], [30] have integrated CNN with Transformer to construct feature complementary networks for various downstream tasks. For example, CMT [9] combined the local features of CNN with the global features of Transformer to provide a high-precision and efficient network. Next-ViT [31] developed a hybrid network of CNN and Transformer, considering latency/accuracy trade-offs across various vision tasks. Mobile-Former [32] established a lightweight image cognition task by bidirectionally integrating CNN and Transformer. ACT [33] utilized a dual-branch fusion of CNN and Transformer to enrich super-resolution recognition tasks.

However, the fused features obtained by the aforementioned methods often fail to accurately represent fasteners and their defects in complex natural scenes, resulting in poor robustness in fastener defect detection. Therefore, strengthening CNN-Transformer feature extraction and fusion to improve the model's representation capability becomes crucial in addressing this challenge. MCHNet-RF2D adopts CNN-Transformer multi-scale feature extraction and multi-scale fusion between local and global features, thereby providing precise representations of fasteners and their defects. Consequently, it effectively resolves fastener defect detection issues in natural real-world scenarios.

III. Methodology

MCHNet-RF2D is designed to enhance the robustness of identifying complex real-world fastener defects. It emphasizes extracting both global semantic features and local detailed features of fasteners and skillfully integrating them to accurately represent fasteners and their defects, thereby improving the precision and recall of defect detection. Global semantic features are utilized for identifying fastener objects, while local detailed features are employed for recognizing various defects in the fasteners.

As depicted in Fig. 5, MCHNet-RF2D consists of three components: the backbone, neck, and head. The backbone leverages a CNN-Transformer hybrid network to extract and alternately fuse local detailed features and global semantic features. The neck performs multi-scale feature fusion between these global and local features, thereby enhancing the effective capture of fastener defect details. Finally, the head utilizes the fused multi-scale features for object detection, encompassing both image classification and object bounding box regression.

The backbone component is divided into four stages, each comprising multiple Efficient CNN Blocks (ECBs) and Multi-scale ViT Blocks (MViTBs). The ECB (Section III-A) extracts local features to enhance defect detail recognition, while the MViTB (Section III-B) extracts global features to enhance fastener semantic recognition. These two types of features are alternately fused to generate the mixed features that possess both the global recognition capability of fasteners and the local detail recognition capability of fastener defects.

PathMerging serves as the connection between the previous and next stages, reducing the resolution by 1/4 from the previous stage while doubling the dimensionality.

A. Efficient CNN Block (ECB)

The efficient CNN block, illustrated in Fig. 6, is designed to extract local features of fasteners in complex real-world scenarios. The ECB achieves superior performance and high computational speed through the incorporation of three supplementary techniques: Group Depth-wise Convolution (GDC) based on channel splitting, Group-wise Attention (GA), and Channel Shuffle (CS).

The implementation of GDC is presented in Formula (1). The feature $\mathbf{X} \in \mathbb{R}^{(W\times H\times C)}$ is first divided into m groups along the channel dimension, with each group G_m having $\frac{C}{m}$ feature channels. Depth-Wise Convolution (DWConv) [34] is then applied within each group to perform efficient feature extraction.

$$\begin{aligned} \mathbf{GDC}(\mathbf{X}) &= \mathrm{DWConv}(\mathrm{Split}(\mathbf{X})) \rightarrow \tilde{\mathrm{G}}_1, \tilde{\mathrm{G}}_2 \ldots \tilde{\mathrm{G}}_\mathrm{m} \\ \mathrm{Split}(\mathbf{X}) &\rightarrow \{\mathrm{G}_1, \mathrm{G}_2 \ldots \mathrm{G}_\mathrm{m}\} \\ \tilde{\mathrm{G}}_\mathrm{k} &= \underbrace{\mathrm{DWConv}(\mathrm{G}_\mathrm{k})}_{k=1,2\ldots m} \end{aligned} \tag{1}$$

To address the varying contributions among GDC groups to the model, we employ Group-wise Attention (GA) to obtain adaptive weights for each group. These weights are then multiplied by the group features to derive each group's representation for the model. The implementation of GA is shown in Equation (2).

$$\begin{aligned} \mathbf{GA}(\tilde{\mathrm{G}}_1, \tilde{\mathrm{G}}_2, \ldots, \tilde{\mathrm{G}}_\mathrm{m}) =& \mathrm{Softmax}(\mathrm{FC}(\mathrm{Swish}(\mathrm{FC}(\\ & \underbrace{\mathrm{Swish}(\mathrm{Conv}(\sum \tilde{\mathrm{G}}_\mathrm{k})))))) \otimes \tilde{\mathrm{G}}_\mathrm{k}}_{\mathrm{k}=1,2\ldots\mathrm{m}} \\ & \rightarrow \hat{\mathrm{G}}_1, \hat{\mathrm{G}}_2 \ldots \hat{\mathrm{G}}_\mathrm{m} \end{aligned} \tag{2}$$

where Swish denotes the swish activation function [35], and FC refers to the fully connected operation.

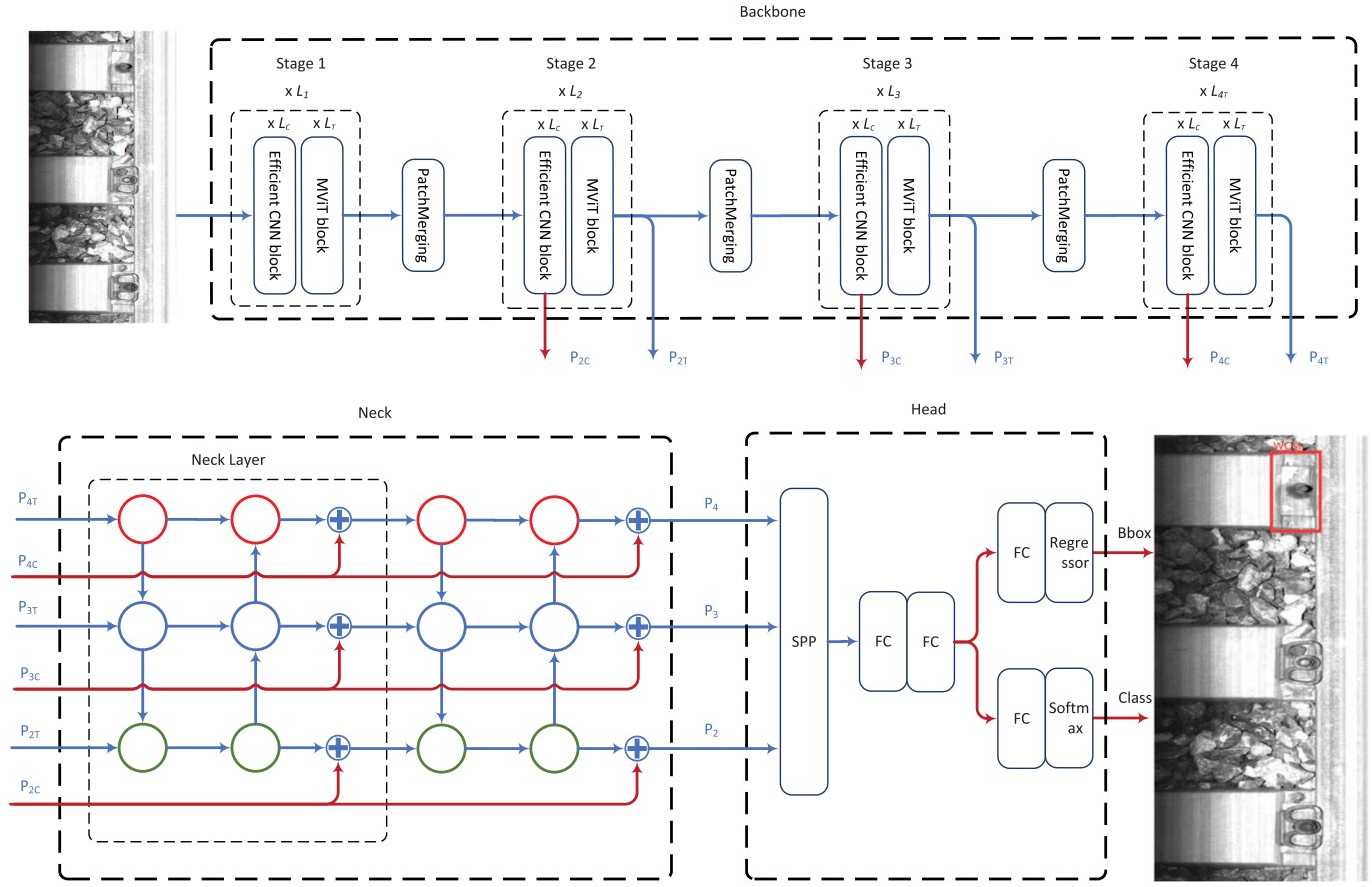


Fig. 5. The overall architecture of MCHNet-RF2D. The efficient CNN block in the backbone is depicted in Fig. 6, and the MViTB is illustrated in Fig. 7.

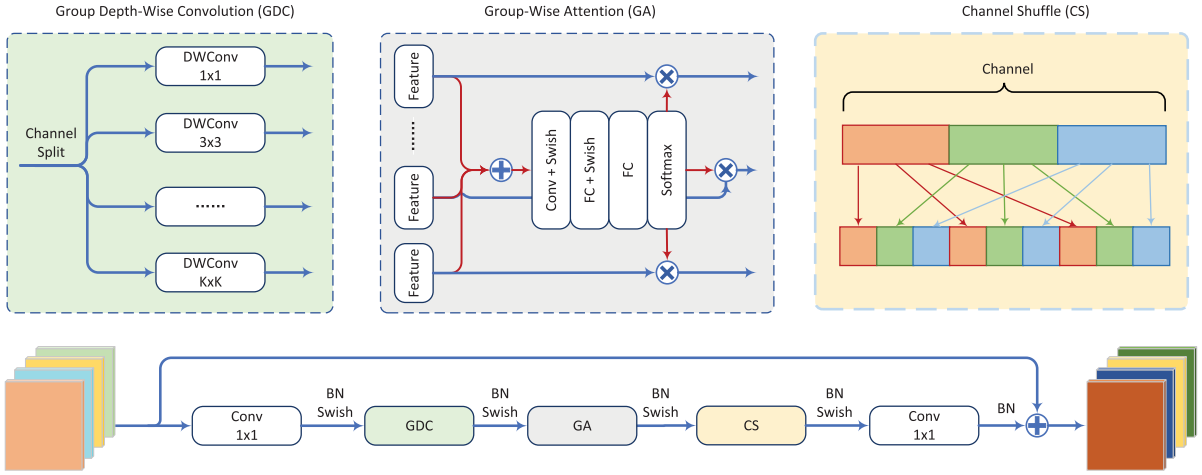


Fig. 6. The ECB architecture primarily consists of three parts: Group Depth-Wise Convolution, Group-wise Attention and channel shuffle.

To facilitate cross-channel communication between $\hat{G}_k$ groups, channel shuffling is employed, as detailed in Formula (3). The shuffled features are then concatenated.

$$\hat{\mathrm{G}}_{\mathrm{k}} = \underbrace{\{\hat{\mathrm{G}}_{\mathrm{k}[1]}, \hat{\mathrm{G}}_{\mathrm{k}[2]} \ldots \hat{\mathrm{G}}_{\mathrm{k}[\frac{\mathrm{C}}{\mathrm{m}}]}\}}_{k=1,2...m}$$

$$\begin{aligned}\mathbf{CS}(\hat{\mathrm{G}}_1, \hat{G}_2, \ldots, \hat{G}_m) = \{&\hat{\mathrm{G}}_{1[1]}, \hat{\mathrm{G}}_{2[1]} \ldots \hat{\mathrm{G}}_{\mathrm{m}[1]}, \\ &\hat{\mathrm{G}}_{1[2]}, \hat{\mathrm{G}}_{2[2]} \ldots \hat{\mathrm{G}}_{\mathrm{m}[2]} \\ &\ldots \\ &\hat{\mathrm{G}}_{1[\frac{\mathrm{C}}{\mathrm{m}}]}, \hat{\mathrm{G}}_{2[\frac{\mathrm{C}}{\mathrm{m}}]} \ldots \hat{\mathrm{G}}_{\mathrm{m}[\frac{\mathrm{C}}{\mathrm{m}}]}\} \rightarrow \hat{\mathbf{X}}\end{aligned} \tag{3}$$

The ECB is implemented as follows:

$$\begin{aligned}\mathbf{ECB}(\mathbf{X}) &= \mathbf{X} + \mathcal{C}_2 \circ \mathcal{CS} \circ \mathcal{GA} \circ \mathcal{GDC} \circ \mathcal{C}_1 \\ \mathcal{C}_1 &= \mathrm{Swish}(\mathrm{BN}(\mathrm{Conv}(\mathbf{X}))) \rightarrow \tilde{\mathbf{X}} \\ \mathcal{GDC} &= \mathbf{GDC}(\tilde{\mathbf{X}}) \rightarrow \tilde{\mathrm{G}}_1, \tilde{\mathrm{G}}_2 \ldots \tilde{\mathrm{G}}_{\mathrm{m}} \\ \mathcal{GA} &= \mathrm{Swish}(\mathrm{BN}(\mathbf{GA}(\tilde{\mathrm{G}}_1, \tilde{\mathrm{G}}_2 \ldots \tilde{\mathrm{G}}_{\mathrm{m}}))) \rightarrow \hat{\mathrm{G}}_1, \hat{\mathrm{G}}_2 \ldots \hat{\mathrm{G}}_{\mathrm{m}} \\ \mathcal{CS} &= \mathrm{Swish}(\mathrm{BN}(\mathbf{CS}(\hat{\mathrm{G}}_1, \hat{\mathrm{G}}_2 \ldots \hat{\mathrm{G}}_{\mathrm{m}})) \rightarrow \hat{\mathbf{X}} \\ \mathcal{C}_2 &= \mathrm{BN}(\mathrm{Conv}(\hat{\mathbf{X}}))\end{aligned} \tag{4}$$

where $\circ$ indicates an execution order from right to left, and BN represents batch normalization.

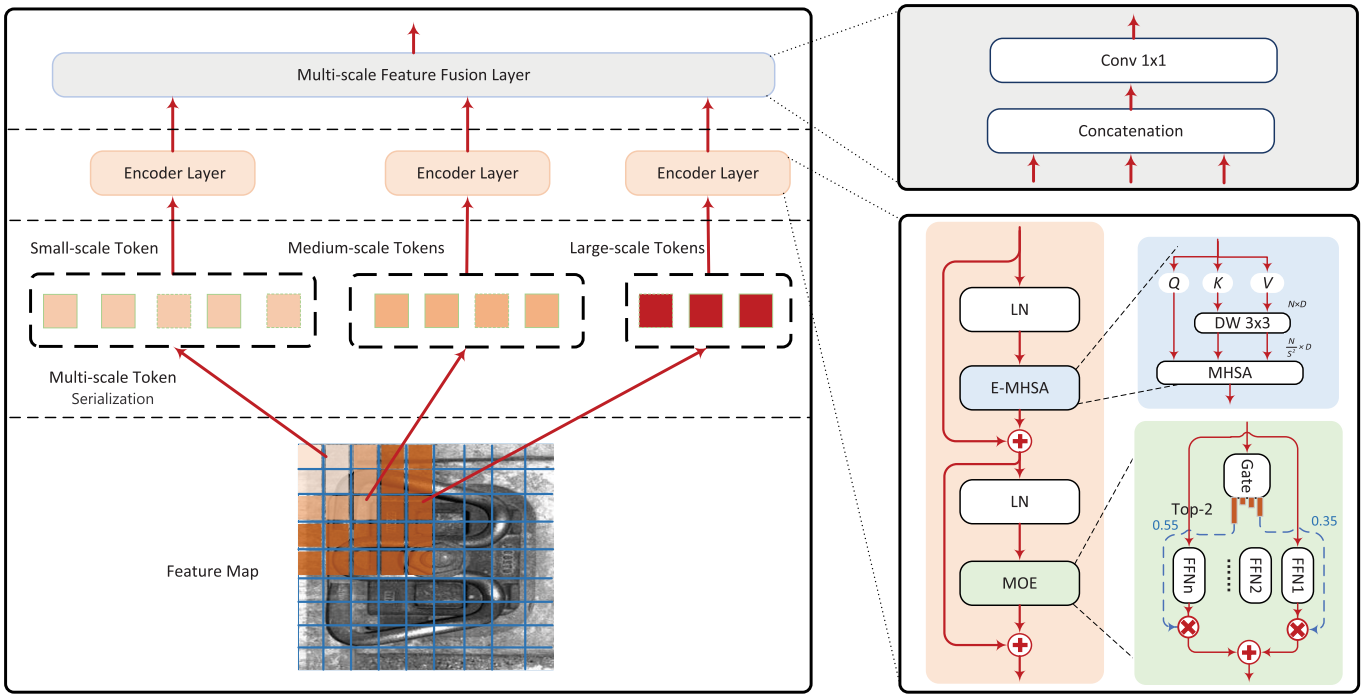


Fig. 7. The MViTB architecture primarily consists of three parts: multi-scale token serialization, encoder layer, and multi-scale feature fusion layer.

B. Mutil-Scale ViT Block (MViTB)

The MViTB, depicted in Fig. 7, extracts global features across multiple scales, capturing information at various levels and granularities. This enhances the model's understanding of image content, and improves its recognition ability of variable scales and occluded fasteners.

MViTB comprises three modules: multi-scale token serialization, encoder layer, and multi-scale feature fusion layer.

1) Multi-Scale Token Serialization (MTS): This module converts the input feature map $\mathbf{F}$ into token embeddings at various scales. MTS employs non-overlapping convolution operation using $K \times K$ convolutional kernels with $K = 2, 3, 5$ to generate small-scale, medium-scale, and large-scale token embeddings, respectively.

$$\mathbf{MTS}(\mathbf{F}) = \mathrm{Conv}(\mathbf{F}, K \times K) \tag{5}$$

2) Encoder Layer: Unlike traditional transformer layers [11], the encoder layer replaces MHSA and FFN with the Efficient Multi-headed Self-attention (E-MHSA) and Mixture Of Experts (MOE) [36], [37], respectively. This modification addresses inefficiency and poor generalization of the traditional transformer layers for global feature extraction, thereby enhancing the efficiency of global feature extraction in the encoder layer and strengthening the model's generalization capabilities.

E-MHSA: The E-MHSA reduces the dimensions of Key $\mathbf{K} \in \mathbb{R}^{(N\times D)}$ and Value $\mathbf{V} \in \mathbb{R}^{(N\times D)}$ through Depth-Wise Convolution (DWConv), improving the efficiency of MHSA matrix dot product. The implementation of E-MHSA is shown in Formula (6).

$$\begin{aligned}
&\mathbf{E\text{-}MHSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{MHSA} \circ \mathrm{Shrink} \\
&\quad \mathrm{Shrink} = \mathrm{DepthConv}(\mathbf{K},\mathbf{V}) \rightarrow \mathbf{K}^*, \mathbf{V}^* \\
&\mathrm{MHSA} = \begin{cases} \underbrace{\mathrm{softmax}\left(\dfrac{\mathbf{Q}^{(m)}\mathbf{K}^{*(m)\top}}{\sqrt{d}}\right) \times \mathbf{V}^{*(m)} \rightarrow \bar{\mathbf{Z}}^{(m)}}_{m=1,2,\ldots M} \\ contact\left(\bar{\mathbf{Z}}^{(1)}, \bar{\mathbf{Z}}^{(2)}, \ldots \bar{\mathbf{Z}}^{(M)}\right) \times W \rightarrow \bar{\mathbf{Z}} \end{cases}
\end{aligned} \tag{6}$$

In this formula, the Shrink operation [9], [38] applies a depth-wise convolution to $\mathbf{K}$ and $\mathbf{V}$, resulting in $\mathbf{K}^* \in \mathbb{R}^{\left(\frac{N}{S^2}\times D\right)}$ and $\mathbf{V}^* \in \mathbb{R}^{\left(\frac{N}{S^2}\times D\right)}$, where S represents the scaling factor. W is a concatenated matrix. The output $\bar{\mathbf{Z}} \in \mathbb{R}^{(N\times D)}$ represents the global feature extracted using E-MHSA.

MOE: Since the MOE adaptively integrates multiple expert networks, each specializing in its area of expertise, enhances the ability to solve complex problems and improves overall prediction accuracy and robustness. Therefore, we introduced MOE to replace the traditional FFN in the transformer to improve the recognition capability of fastener defects in complex natural scenarios.

The MOE comprises a gate network and N feed-forward networks ($\mathrm{FFN}_1 \ldots \mathrm{FFN}_n$), where each FFN network serves as an expert network. Given a feature $\mathbf{Z} \in \mathbb{R}^{(M\times D)}$, where M represents the number of tokens and D represents the token dimension, each token is represented by $\mathbf{Z}_k$, where $k = 1, 2 \ldots M$. The implementation of MOE can be

found as follows:

$$\begin{aligned}\mathbf{MOE}(\mathbf{Z}) &= \underbrace{\sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{G_i}(\mathrm{Z}_k)\mathrm{E_i}(\mathrm{Z}_k)}_{k=1,2...M} \\ G_i(\mathrm{Z}_k) &= \mathrm{Softmax}(\mathrm{Top2}(\mathrm{Linear}(\mathrm{Z}_k))) \\ \mathrm{Linear}(\mathrm{Z}_k) &= \mathrm{Z}_k \in \mathbb{R}^{(1\times \mathrm{D})} \rightarrow \tilde{\mathrm{Z}}_k \in \mathbb{R}^{(1\times \mathrm{N})}\end{aligned} \tag{7}$$

Here, $\mathrm{E}_i(\cdot)$ represents the result of the i-th selected expert network (FFN_i). Unselected experts do not perform their computations, setting their $\mathrm{E}_i(\cdot)$ to 0. Each expert network consists of a two-layer fully connected network. For each token $\mathbf{Z}_k$, the gate network $\mathrm{G}_i(\cdot)$ determines the contribution of each expert network, selecting only the top two expert networks with the highest contributions. These selected expert networks are then multiplied by their respective contributions, and the results are summed to obtain the MOE output.

3) Multi-Scale Feature Fusion Layer (M2FL): This layer combines multi-scale encoded features via element-wise concatenate, followed by a 1 × 1 convolution.

$$\mathbf{M2FL}(\mathbf{Z_S}, \mathbf{Z_M}, \mathbf{Z_L}) = \quad \mathrm{BN}(\mathrm{Conv}(\mathrm{contact}(\mathbf{Z_S}, \mathbf{Z_M}, \mathbf{Z_L}))) \tag{8}$$

where $\mathbf{Z_S}$, $\mathbf{Z_M}$, and $\mathbf{Z_L}$ represent the features encoded at small, medium, and large scales, respectively.

The implementation of MViTB is as follows:

$$\begin{aligned}\mathbf{MViTB}(\mathbf{Z}) &= \mathcal{M}\mathrm{2FL} \circ \mathcal{E}\mathrm{ncoder} \circ \mathcal{M}\mathrm{TS} \\ \mathcal{M}\mathrm{TS} &= \mathbf{MTS}(\mathbf{Z}) \rightarrow \bar{\mathbf{Z}}_\mathbf{S}, \bar{\mathbf{Z}}_\mathbf{M}, \bar{\mathbf{Z}}_\mathbf{L} \\ \mathcal{E}\mathrm{ncoder} &= \underbrace{\mathbf{MOE}(\mathrm{LN}(\mathrm{Z} + \mathrm{LN}(\mathbf{E\text{-}MHSA}(\bar{\mathbf{Z}}_\mathbf{n}))))}_{n=S,M,L} \\ &\rightarrow \mathbf{Z_S}, \mathbf{Z_M}, \mathbf{Z_L} \\ \mathcal{M}\mathrm{2FL} &= \mathbf{M2FL}(\mathbf{Z_S}, \mathbf{Z_M}, \mathbf{Z_L})\end{aligned} \tag{9}$$

C. Neck

Drawing inspiration from FPN [25], PaNet [26], and biFPN [39], we integrate the semantic information of high-level features (e.g., P_4) with the detailed information of low-level features (e.g., P_2) in the neck component. This enables the MCHNet-RF2D network to simultaneously acquire rich semantic and detailed information for object detection, thereby enhancing its robustness and generalization capability.

Bidirectional fusion methods (top-down and bottom-up) are regarded as a single neck layer. In the top-down approach, we employ upsampling, while in the bottom-up approach, we utilize downsampling. Since fastener defect detection relies on local details, we further fuse the result with local features after the bottom-up fusion. The specific implementation of the neck layer is as follows:

$$\begin{aligned}&\mathbf{NeckLayer}(\mathbf{P_{4T}}, \mathbf{P_{4C}}, \mathbf{P_{3T}}, \mathbf{P_{3C}}, \mathbf{P_{2T}}, \mathbf{P_{2C}}) \\ &\quad = \mathbf{P''_{4T}}, \mathbf{P''_{3T}}, \mathbf{P''_{2T}} \\ &\mathbf{P''_{2T}} = \mathrm{P_{2C}} + \mathbf{P'_{2T}} \\ &\mathbf{P''_{3T}} = \mathrm{P_{3C}} + \mathbf{P'_{3T}} \\ &\mathbf{P''_{4T}} = \mathrm{P_{4C}} + \mathbf{P'_{4T}} \\ &\mathbf{P'_{2T}} = \mathrm{P_{2T}} + \mathrm{UpSampling}(\mathrm{P_{3T}} \\ &\qquad + \mathrm{UpSampling}(\mathrm{P_{4T}})) \\ &\mathbf{P'_{3T}} = \mathrm{P_{3T}} + \mathrm{UpSampling}(\mathrm{P_{4T}}) \\ &\qquad + \mathrm{DownSampling}(\mathrm{P'_{2T}}) \\ &\mathbf{P'_{4T}} = \mathrm{P_{4T}} + \mathrm{DownSampling}(\mathrm{P'_{3T}})\end{aligned} \tag{10}$$

where $\mathrm{P'_{2T}}$, $\mathrm{P'_{3T}}$, and $\mathrm{P'_{4T}}$ respectively represent the outputs of each layer after bottom-up fusion.

To ensure a more thorough and effective fusion of high-level and low-level features in the neck, we further optimize by iterating the neck layer multiple times, as shown in Formula (11).

$$\begin{aligned}&\mathbf{Neck}(\mathbf{P_{4T}}, \mathbf{P_{4C}}, \mathbf{P_{3T}}, \mathbf{P_{3C}}, \mathbf{P_{2T}}, \mathbf{P_{2C}}) \\ &\quad = \underbrace{\mathbf{NeckLayer}(\mathbf{P_{4T}}, \mathbf{P_{4C}}, \mathbf{P_{3T}}, \mathbf{P_{3C}}, \mathbf{P_{2T}}, \mathbf{P_{2C}})}_{\times \mathrm{N}} \\ &\quad = \mathbf{P_4}, \mathbf{P_3}, \mathbf{P_2}\end{aligned} \tag{11}$$

D. Head

The head component is responsible for precise defect bounding box localization and classification. The implementation of the head is as shown in Formula (12). Initially, Spatial Pyramid Pooling [40] (SPP) transforms the scale-invariant features of P_2, P_3, and P_4 using 1 × 1, 2 × 2, and 4 × 4 max-pooling, respectively, to generate fixed-length feature representations. These representations are then fed into fully connected (FC) networks for defect classification and localization

$$\begin{aligned}\mathbf{Head}(\mathbf{P_4}, \mathbf{P_3}, \mathbf{P_2}) &= \mathcal{B}\mathrm{box\text{-}Class} \circ \mathcal{FC} \circ \mathcal{SPP} \\ \mathcal{SPP} &= \begin{cases} \mathrm{MAXPooling1} \times 1(\mathbf{P_4}) \rightarrow \mathbf{P'_4} \\ \mathrm{MAXPooling2} \times 2(\mathbf{P_3}) \rightarrow \mathbf{P'_3} \\ \mathrm{MAXPooling4} \times 4(\mathbf{P_2}) \rightarrow \mathbf{P'_2} \\ \mathrm{Contact}(\mathbf{P'_4} + \mathbf{P'_3} + \mathbf{P'_2}) \rightarrow \mathrm{P} \end{cases} \\ \mathcal{FC} &= \underbrace{\mathrm{BN}(\mathrm{FC}(\mathrm{P}))}_{\times 2} \rightarrow \mathrm{P'} \\ \mathcal{B}\mathrm{box\text{-}Class} &= \begin{cases} \mathrm{Regression}(\mathrm{FC}(\mathrm{P'})) \rightarrow \mathrm{Bbox} \\ \mathrm{Softmax}(\mathrm{FC}(\mathrm{P'})) \rightarrow \mathrm{Class} \end{cases}\end{aligned} \tag{12}$$

E. Network Specification

To strike a balance between recognition performance and efficiency, we provide the specific configurations of MCHNet-RF2D in Table I. Notably, the iteration count for the four stages in the backbone is set to 1, 1, 3, and 1, respectively. Within each stage, the ECB iterates 5 times, while the MViTB iterates 2 times. Furthermore, the neck layer in the neck undergoes 2 iterations. The token dimension of MViTBs in the four stages is outlined in the last three columns of Table I.

IV. Experiment

A. Dataset, Training and Evaluation Metrics

1) Dataset: Rail fastener defect samples were collected from over 30 Chinese high-speed trains and urban subway systems. These dataset consists of four categories of defects, with the number of samples in each category detailed in Table II. It is divided into three subsets: the training set (80% of the total samples), the validation set (5%), and the test set (15%).

TABLE I
PARAMETER CONFIGURATION OF M2CTFNET

MCHNet-RF2D	L_1, L_2 L_3, L_4	L_c	L_T	Iteration neck layer	Token dimension
Config	1, 1, 3, 1	5	2	2	[128, 256, 512, 1024]

TABLE II
NUMBER OF SAMPLES FOR EACH CATEGORY OF FASTENER DEFECT DATASET

Category	Number
ESD	320
ECM	830
WCD	1260
WCM	310

2) Training: The MCHNet-RF2D network is pre-trained for 300 epochs on the Image-22K dataset [41], using the AdamW optimizer [42] with a batch size of 256 and a weight decay of 0.05. Next, the network is fine-tuned using the fastener defect dataset for 36 epochs, with a momentum of 0.9 and a weight decay of 0.0005. All networks are trained on 8 Titan RTX GPUs, each with 24GB of memory. To address the challenges posed by imbalanced and difficult samples, we employ the focal loss algorithm [43] as the model's loss function.

3) Evaluation Metrics: We evaluate the performance of the MCHNet-RF2D using metrics such as top-1 accuracy [44], recall [44], mean Average Precision (mAP) [45].

B. Comparison With State-of-the-Art Networks

1) Experimental Setup: To assess the performance of MCHNet-RF2D, we conduct comparisons with four state-of-the-art object detection methods: GDD models,[1] CNN-based models, ViT-based models, and CNN/ViT hybrid (C/V) models. Evaluation metrics encompass mAP and recall.

2) Object Detection Result: The experimental results, shown in Table III, indicate that MCHNet-RF2D achieves an impressive mAP of 94.9 while maintaining lower computational costs and fewer weight parameters. Specifically, compared to the CNN-based models, MCHNet-RF2D outperforms the YOLOv7 model by 3.2%. When compared to the ViT-based models, it exceeds the Dual-ViT-S model by 2.6%. In comparison with the GDD models, it surpasses the InCTRL model by 3.2%. When benchmarked against the C/V models, it outperforms the ACT model by 2.8%.

3) Recall: Taking MCD defect detection as an example, MCHNet-RF2D surpasses YOLOv7, Dual-ViT-S, InCTRL, and ACT by 4.6%, 3.5%, 4.1%, and 3.9%, respectively.

MCHNet-RF2D demonstrates outstanding performance with low computational costs and fewer weight parameters, primarily due to two main reasons: 1) By employing the MViTB for extracting global features and the ECB for extracting local features, MCHNet-RF2D enriches feature extraction and improves the robustness of fastener defect recognition under complex real-world conditions. 2) MCHNet-RF2D effectively integrates global and local features at multiple scales, creating complementary advantages and enhancing fastener detection accuracy and recall.

[1]Due to the limitation of using 8 or 16-shot validation in GDD models for fastener defect detection, resulting in subpar performance, we fine-tuned the GDD model using our fastener training set and subsequently validated it with the test set.

C. Feature Visualization Comparison

1) Experimental Setup: To evaluate the capability of MCHNet-RF2D in extracting features related to fastener defects, we use the Faster R-CNN network [45] as the baseline and then replace its backbone with those from YOLOv7 [47], Dual-ViT-S [51], InCTRL [15], ACT [33], and our M2CTFNet. The features extracted by these five networks are then visualized and compared.

2) Result: The experimental results, shown in Fig. 8, indicate that all the networks are capable of extracting features of rail fastener defects, our M2CTFNet not only covers the entire fastener but also focuses specifically on defect areas. This ability aids in defect identification. These findings demonstrate that M2CTFNet excels in accurately capturing the overall structure of fasteners and precisely identifying areas with potential defects. Furthermore, M2CTFNet proves to be particularly effective in extracting features in complex real-world scenarios.

D. Seeking Optimal Parameters L_C and L_T

We set the iteration times for the four stages of the MCHNet-RF2D's backbone to 1, keeping $L_T = 1$ and $L_C = 1$ fixed for Stages 1, 2, and 4. We then vary the L_T and L_C iteration parameters for Stage 3 to determine the optimal configuration. Utilizing the MCHNet-RF2D's backbone as the backbone for our experimental classification network, we combine different L_T and L_C parameters as detailed in Table IV to identify the optimal values.

The experimental findings, detailed in Table IV, indicate that with L_T parameters fixed in Stage 3, the performance of MCHNet-RF2D steadily improves with increasing L_C iteration counts. However, when L_C exceeds the threshold of 5, further increases in L_C iteration count do not enhance model performance and instead increase weight parameters and computational overhead.

Conversely, keeping the L_C parameter constant, increasing the L_T iteration count also improves the performance of MCHNet-RF2D. Nevertheless, once L_T counts exceed 2, the model's performance plateaus, accompanied by an increase in overhead.

Therefore, with $L_T = 2$ and $L_C = 5$, MCHNet-RF2D achieves an optimal balance between performance and overhead.

E. Impact of L_3 Iteration Counts on Performance

With $L_T = 2$ and $L_C = 5$, we aim to find the optimal L_3 parameters for Stage 3. Inspired by ResNet's approach of increasing block numbers in Stage 3 [56], we set the iteration

TABLE III
COMPARING MCHNET-RF2D WITH THE EXISTING STATE-OF-THE-ART NETWORKS

Mode	Backbone	#param (M)	FLOPS (B)	mAP AP^{box}	Recall ESD	Recall ECM	Recall WCD	Recall WCM
CNN	ResNet-101	63.2	336	88.4	87.3	88.5	90.0	86.8
	DCNN [8] *	68.2	356	89.2	87.5	88.9	90.5	87.0
	Su *et al.* [6] *	-	-	89.4	87.9	89.0	90.9	87.2
	EfficientDet-B6 [27]	52.1	226	91.0	90.3	91.4	92.5	89.4
	Dynamic Head [46]	56.7	277	91.6	90.5	91.8	92.9	89.6
	YOLOv7 [47]	54.0	332	91.7	90.7	91.9	93.1	89.8
ViT	PVT-S [38]	44.1	245	87.3	87.0	88.2	89.7	86.2
	Swin-T [48]	47.8	264	88.2	87.4	88.9	89.8	86.6
	DAT-T [49]	48.5	272	89.5	88.5	89.6	90.9	87.1
	Twins-S [50]	44.3	245	89.6	88.7	90.3	91.2	87.8
	MPViT-S [24]	43.5	268	91.7	90.5	91.6	92.9	89.8
	Dual-ViT-S [51]	46.7	277	92.3	91.1	92.4	94.2	90.4
GDD	Tip-Adapter [52]	196.3	392	89.4	88.9	89.5	91.0	87.4
	WinCLIP [14]	204.4	396	90.6	89.6	90.8	91.3	88.9
	IM-IAd [17]	211.5	393	91.5	90.4	91.4	92.1	89.6
	InCTRL [15]	222.0	415	91.7	90.8	92.5	93.6	89.1
C/V	MF-508M[32]	17.9	168	85.1	83.2	85.8	87.1	83.1
	Conformer-S [53]	53.3	346	91.6	90.9	91.7	92.6	89.8
	YOLOv8-FAM [5] *	68.0	475	92.0	90.8	92.4	93.1	90.2
	CTCNet [54]	56.9	458	92.0	91.0	92.4	93.2	90.0
	ACT [33]	54.2	354	92.1	91.3	92.7	93.8	90.5
Our	**M2CTFNet (our)**	48.3	295	**94.9**	**93.8**	**95.3**	**97.7**	**92.6**

* These algorithms are specifically designed to detect rail fastener defects.

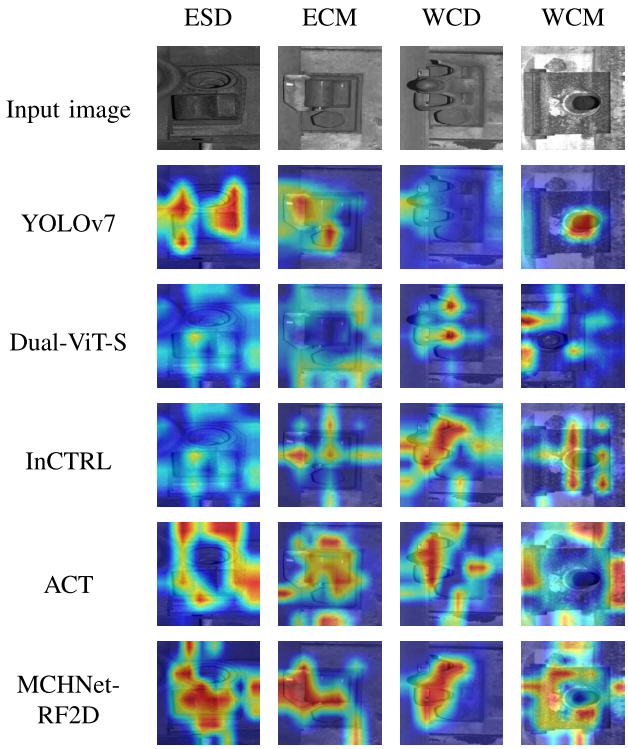


Fig. 8. Grad-CAM++ [55] visualization results. We compare the visualization results of YOLOv7 [47], Dual-ViT-S [51], InCTRL [15], ACT [33] in object detection. The Grad-CAM++ visualization is calculated for the last convolutional outputs.

counts for Stages 1-2 and 4 (denoted as L_1, L_2, and L_4) to 1. Subsequently, we vary the iteration count of L_3, as illustrated in Table V. At L_3 = 3, MCHNet-RF2D achieves an optimal balance between performance and overhead costs. Further increases in L_3 do not enhance performance and instead escalate overhead costs.

TABLE IV
TO IDENTIFY THE OPTIMAL L_T AND L_C PARAMETERS IN THE BACKBONE, WE HOLD THE ITERATION COUNTS CONSTANT FOR STAGES 1-4 AND ALSO FIX THE L_T AND L_C ITERATION COUNTS IN STAGES 1-2 AND 4. BY VARYING THE L_T AND L_C ITERATION COUNTS IN STAGE 3 OF THE CLASSIFICATION NETWORK, WE DETERMINE THE OPTIMAL PARAMETERS

Stage 3 L_T	Stage 3 L_C	#param (M)	FLOPS (B)	ACC TOP-1
1	3	4.1	28.2	91.1
	4	4.2	28.5	91.3
	5	4.3	28.9	91.6
	6	4.4	29.3	91.6
	7	4.6	29.7	91.5
2	3	6.3	32.5	91.7
	4	6.4	33.1	92.0
	5	6.5	33.7	**92.2**
	6	6.7	34.3	92.2
	7	6.8	35.0	92.0
3	3	9.7	37.0	91.6
	4	9.8	37.6	91.9
	5	9.9	38.3	92.1
	6	10.1	38.8	92.1
	7	10.2	39.5	92.0

TABLE V

FIXING THE ITERATION COUNTS OF STAGES 1-2 AND 4, THE ITERATION COUNT OF STAGE 3 IS VARIED TO FIND THE OPTIMAL BALANCE PARAMETER L_3

Stage 3			#param (M)	FLOPS (B)	ACC
L_T	L_C	L_3			TOP-1
2	5	1	6.5	33.9	92.2
		2	10.2	54.5	93.4
		3	16.3	80.3	**94.1**
		4	24.1	118.9	94.0
		5	32.8	132.1	93.8

TABLE VI

DISPLAYING THE CORRELATION BETWEEN NECK LAYER ITERATION COUNTS AND MODEL PERFORMANCE

Neck Layer Iteration Times	#param (M)	FLOPS (B)	ACC Top-1
1	31.6	160.2	94.1
2	32.1	161.5	**95.0**
3	32.8	163.2	95.1

TABLE VII

ABLATION EXPERIMENT ON THE EFFECTIVENESS OF CHANNEL SHUFFLE, AND GROUP-WISE ATTENTION IN OUR MCHNET-RF2D

+ Channel Shuffle (CS)	+ Group-wise Attention (GA)	#param (M)	FLOPS (B)	ACC Top-1
×	×	101	63.0	95.0
✓	×	+0.1	+0.2	95.2
×	✓	+0.7	+1.2	95.4
✓	✓	+0.9	+1.5	**95.6**

F. Impact of Neck Layer Iteration Times on Performance

In MCHNet-RF2D, the neck comprises multiple neck layers. Table VI illustrates the impact of neck layer iteration times on model performance. Optimal performance and overhead balance are achieved when the iteration count is 2. Further increases in iteration count yield slight performance improvements, but lead to a continuous increase in weight parameters. Hence, MCHNet-RF2D adopts neck layers with 2 iterations.

G. Ablation Studies

1) Impact of Channel Shuffle and Channel-Wise Attention on Model Performance: To evaluate the impact of channel shuffle and channel-wise attention modules in the ECB on MCHNet-RF2D, we construct a baseline classification network by removing these two modules from the MCHNet-RF2D architecture. Subsequently, we reintroduce the channel shuffle and group-wise attention modules one by one to this baseline network to assess their effects on MCHNet-RF2D's performance.

Table VII summarizes our experimental results. The channel shuffle module improves MCHNet-RF2D's performance by 0.2%, while the group-wise attention module yields a 0.4% improvement. Combining both modules results in a slight increase in computational overhead and weight parameters, but the overall performance improvement reaches 0.6%.

TABLE VIII

ABLATION EXPERIMENT ON DIFFERENT SCALES OF THE MVITB COMPONENT IN THE MCHNET-RF2D

Submodule		MCHNet-RF2D						
MViTB	Small-scale	w/	w/o	w/o	w/	w/	w/o	w/
	Medium-scale	w/o	w/	w/o	w/	w/o	w/	w/
	Large-scale	w/o	w/o	w/	w/o	w/	w/	w/
#param (M)		33.0	32.4	32.2	34.0	33.7	33.2	36.2
FLOPS (B)		163	161	158	170	168	165	185
ACC Top-1		96.0	95.8	95.7	96.4	96.6	96.3	**96.8**

TABLE IX

ABLATION EXPERIMENT ON MOE IN THE MCHNET-RF2D

	#param (M)	FLOPS (B)	Det (mAP)
MOE	36.4	191	**97.7**
FFN	34.7	187	96.9

2) Impact of MViTB on Model Performance: The MViTB component incorporates multi-scale token encoders, and we assess their impact from three perspectives: 1) the impact of each scale on the performance of MCHNet-RF2D; 2) the effect of combining any two scales; and 3) the effect of combining all three scales.

Table VIII presents our experimental findings. Token encoders at small, medium, and large scales enhance MCHNet-RF2D's performance by 0.5%, 0.3%, and 0.2%, respectively. Combining any two scales also yields performance gains, with the small and large-scale combination notably improving by 1.1%. When all three scales are combined, the performance increases by 1.4%. Despite the associated increase in computational overhead and weight parameters, the performance improvements justify this combination.

3) MOE Vs. FFN in the Encoder Layer: To assess the impact of the MOE module on MCHNet-RF2D's performance, we create a new classification network by replacing MOE with FFN in the MCHNet-RF2D. This modified network is then compared against the original MCHNet-RF2D to assess the difference.

Table IX shows the comparison results. Using MOE instead of FFN improves model performance by 0.8%. Although using MOE incurs a higher computational cost and requires more weight parameters compared to FFN, the significant performance enhancement justifies its utilization in the model.

H. Results of Fasteners Defect Recognition Under Various Real-World Scenarios

MCHNet-RF2D has been deployed on over 40 Chinese high-speed trains and urban subway systems, where it has successfully detected more than 2,000 rail fastener defects. Some detection results are shown in Fig. 9 and Fig. 10.

MCHNet-RF2D exhibits precise detection capabilities not only in clear rail inspection images but also in complex scenarios. These scenarios include instances where cameras

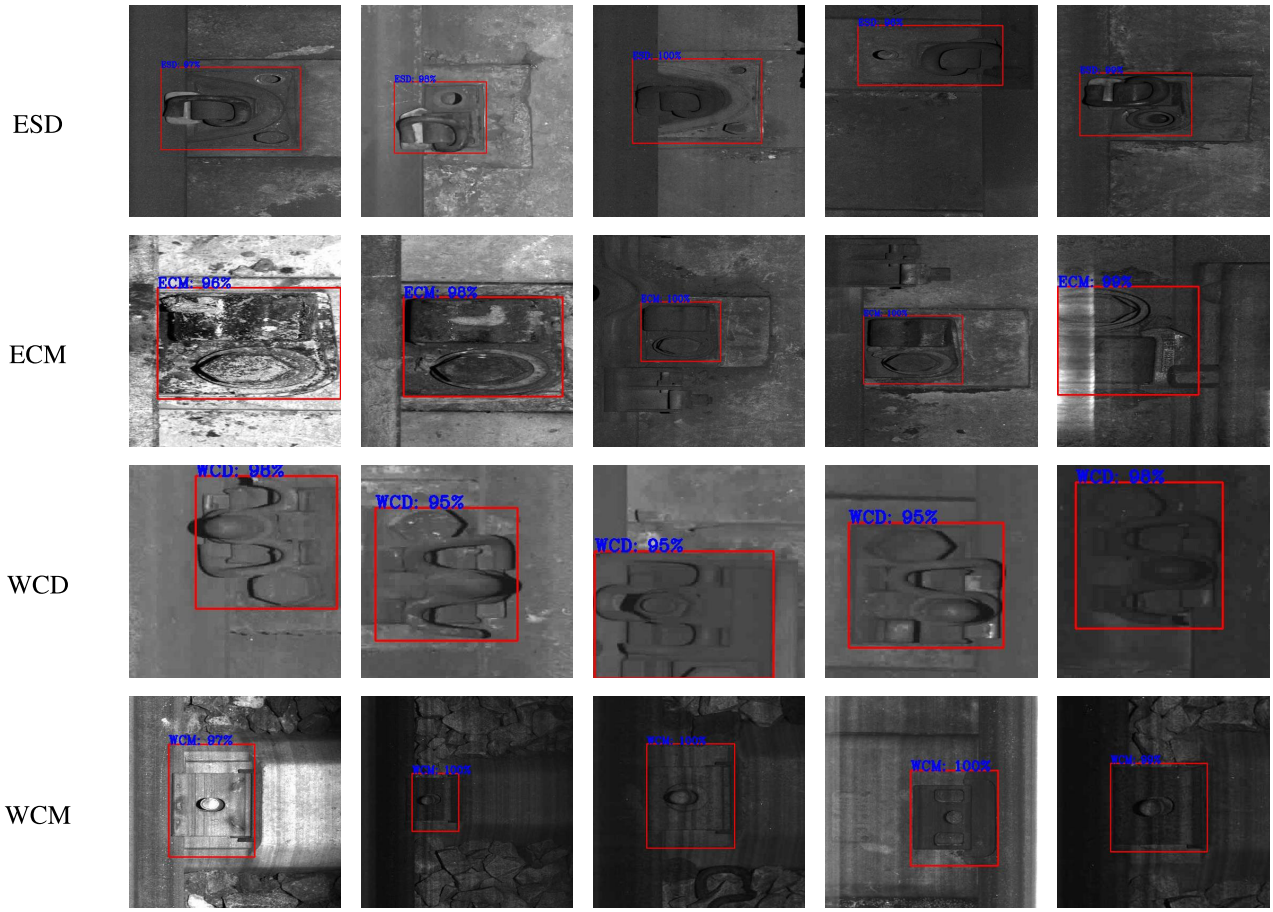


Fig. 9. Partial results of using the CTBM-DAHD model to recognize four categories of fastener defects.

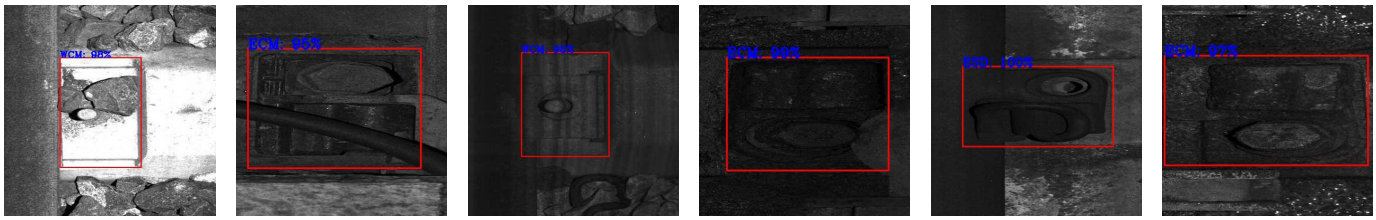


Fig. 10. The MCHNet-RF2D model demonstrates robust recognition capabilities for fastener defects in intricate real-world scenes. In (a)(b), the model is capable of accurately identifying fastener defects in images affected by stones and cables occlusion. In (c), the model adeptly discerns fastener defects in a blurred image. In scenarios where there's insufficient lighting in the image (d), fasteners are corroded (e), and fasteners are submerged in water (f), the model can still precisely identify fastener defects despite such complex conditions.

malfunction (second row, fifth column in Fig. 9), fasteners are obstructed (Fig. 10(a) and Fig. 10(b)), images are blurry (Fig. 10(c)), lighting is insufficient (Fig. 10(d)), fasteners are rusty ((Fig. 10(e)), and fasteners are submerged in water (Fig. 10(f)).

These results demonstrate that MCHNet-RF2D is an effective algorithm for industrial fastener defect detection in complex real-world conditions.

V. Discussion

To address the challenges of identifying diverse fastener defects in high-speed railways and subways, as well as improving detection accuracy in complex natural scenarios, we propose a multi-scale CNN-Transformer hybrid network tailored to these issues. Compared with existing fastener defect detection networks, the state-of-the-art GDD network, and CNN-Transformer models, MCHNet-RF2D demonstrates superior performance in metrics such as mAP and recall, while maintaining lower weight parameters and computational costs.

Nevertheless, MCHNet-RF2D has certain limitations:

- In scarce scenarios, such as when fasteners are completely submerged in water (rendering them indistinguishable) or when captured images are entirely dark (making fasteners invisible), the network is unable to detect defects.
- Although the algorithm has been deployed on more than 40 vehicles and has successfully detected over 2,000 fastener defects, it has not yet been fully validated across all

railway lines in China, leaving certain special scenarios unexamined.

Future work will focus on addressing these limitations by simulating exceptional scenarios and further optimizing the MCHNet-RF2D algorithm to ensure its applicability across all conditions.

In addition to fastener defect detection, MCHNet-RF2D has been applied to defect detection in high-speed railway catenary components, such as loosened or broken droppers and insulator defects, where it has also demonstrated exceptional performance.

VI. Conclusion

In this paper, we propose MCHNet-RF2D, an innovative multi-scale CNN-Transformer hybrid network tailored for detecting rail fastener defects under challenging real-world environments. MCHNet-RF2D leverages the ECB module to extract local detailed features and the MViTB module to capture global semantic information across multiple scales. These features are seamlessly integrated through two efficient fusion stages: alternating fusion in the backbone and multi-scale fusion in the neck. This enhancement significantly boosts the robustness of MCHNet-RF2D, leading to improvements in both the recall and precision of fastener defect recognition. Extensive experiments and real-world deployment of MCHNet-RF2D validate its substantial efficacy and practical applicability. Our research not only offers valuable insights but also opens new avenues for advancing industrial defect detection in complex real-world scenarios.

References

[1] H. Zhang et al., "MRSDI-CNN: Multi-model rail surface defect inspection system based on convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 11162–11177, Aug. 2022.

[2] X. Ni, H. Liu, Z. Ma, C. Wang, and J. Liu, "Detection for rail surface defects via partitioned edge feature," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5806–5822, Jun. 2022.

[3] Z. Chen, Q. Wang, Q. He, T. Yu, M. Zhang, and P. Wang, "CUFuse: Camera and ultrasound data fusion for rail defect detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21971–21983, Nov. 2022.

[4] J. Liu, Z. Ma, Y. Qiu, X. Ni, B. Shi, and H. Liu, "Four discriminator cycle-consistent adversarial network for improving railway defective fastener inspection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10636–10645, Aug. 2022.

[5] S. Qiu et al., "Automated detection of railway defective fasteners based on YOLOv8-FAM and synthetic data using style transfer," *Autom. Construction*, vol. 162, Jun. 2024, Art. no. 105363.

[6] S. Su, S. Du, and X. Lu, "Geometric constraint and image inpainting-based railway track fastener sample generation for improving defect inspection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23883–23895, Dec. 2022.

[7] L. Xiao, B. Wu, and Y. Hu, "Missing small fastener detection using deep learning," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[8] J. Liu, H. Liu, C. Chakraborty, K. Yu, X. Shao, and Z. Ma, "Cascade learning embedded vision inspection of rail fastener by using a fault detection IoT vehicle," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3006–3017, Feb. 2023.

[9] J. Guo et al., "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12175–12185.

[10] Z. Dai et al., "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 3965–3977.

[11] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[12] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[13] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 8748–8763.

[14] J. Jeong, Y. Zou, T. Kim, D. Zhang, A. Ravichandran, and O. Dabeer, "WinCLIP: Zero-/few-shot anomaly classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19606–19616.

[15] J. Zhu and G. Pang, "Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 17826–17836.

[16] G. Xie, J. Wang, J. Liu, F. Zheng, and Y. Jin, "Pushing the limits of fewshot anomaly detection in industry vision: Graphcore," 2023, *arXiv:2301.12082*.

[17] G. Xie et al., "IM-IAD: Industrial image anomaly detection benchmark in manufacturing," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 2720–2733, May 2024.

[18] T. D. Tien et al., "Revisiting reverse distillation for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24511–24520.

[19] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD—A comprehensive real-world dataset for unsupervised anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9584–9592.

[20] Y. Zou, J. Jeong, L. Pemula, D. Zhang, and O. Dabeer, "Spot-the-difference self-supervised pre-training for anomaly detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 392–408.

[21] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "EfficientViT: Memory efficient vision transformer with cascaded group attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14420–14430.

[22] H. Fan et al., "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6824–6835.

[23] C.-F. R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2021, pp. 357–366.

[24] Y. Lee, J. Kim, J. Willette, and S. J. Hwang, "MPViT: Multi-path vision transformer for dense prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7287–7296.

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

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

[27] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[28] J. Pan et al., "EdgeViTS: Competing light-weight CNNS on mobile devices with vision transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 294–311.

[29] J. He, R. Duan, M. Dong, Y. Kao, G. Guo, and J. Liu, "CNN-transformer bridge mode for detecting arcing horn defects in railway sectional insulator," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–16, 2024.

[30] J. Liu et al., "Deep industrial image anomaly detection: A survey," *Mach. Intell. Res.*, vol. 21, no. 1, pp. 104–135, Feb. 2024.

[31] J. Li et al., "Next-ViT: Next generation vision transformer for efficient deployment in realistic industrial scenarios," 2022, *arXiv:2207.05501*.

[32] Y. Chen et al., "Mobile-former: Bridging mobilenet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 5270–5279.

[33] J. Yoo, T. Kim, S. Lee, S. H. Kim, H. Lee, and T. H. Kim, "Enriched CNN-transformer feature aggregation networks for super-resolution," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Oct. 2023, pp. 4956–4965.

[34] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[35] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017, *arXiv:1710.05941*.

[36] D. Lepikhin et al., "GShard: Scaling giant models with conditional computation and automatic sharding," 2020, *arXiv:2006.16668*.

[37] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res.*, vol. 23, no. 120, pp. 1–39, 2022.

[38] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 568–578.

[39] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[42] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

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

[45] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 91–99.

[46] X. Dai et al., "Dynamic head: Unifying object detection heads with attentions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7373–7382.

[47] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[48] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[49] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4794–4803.

[50] X. Chu et al., "Twins: Revisiting the design of spatial attention in vision transformers," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9355–9366.

[51] T. Yao, Y. Li, Y. Pan, Y. Wang, X.-P. Zhang, and T. Mei, "Dual vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 10870–10882, Sep. 2023.

[52] R. Zhang et al., "Tip-adapter: Training-free adaption of clip for few-shot classification," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 493–510.

[53] Z. Peng et al., "Conformer: Local features coupling global representations for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 367–376.

[54] G. Gao, Z. Xu, J. Li, J. Yang, T. Zeng, and G.-J. Qi, "CTCNet: A CNN-transformer cooperation network for face image super-resolution," *IEEE Trans. Image Process.*, vol. 32, pp. 1978–1991, 2023.

[55] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.

[56] F. Wang et al., "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.

Jin He received the M.S. and Ph.D. degrees in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2005 and 2016, respectively. He was a Senior Engineer with Huawei, Shenzhen, China, from 2005 to 2008. He leaded two teams (unified threat management team and Web application firewall team) as the Director of research and development at Link-Trust Company Ltd., Beijing, China, from 2008 to 2011. He is currently working with the School of Automation, Chengdu University of Information Engineering. His research interests include deep learning, high-speed railway defect detection, virtualization, operating systems, and cloud security.

Wei Wang received the Ph.D. degree in computer science and technology from Sichuan University. He is currently a Lecturer at the School of Automation, Chengdu University of Information Technology. He has participated in more than ten scientific research projects, including the National Defense Basic Research Nuclear Science Challenge Special Research Project, the Key Laboratory of Confidential Communication National Defense Technology Project, the National Natural Science Foundation General Project, and Sichuan Province Key Research and Development Project. Currently, he has 14 publications in several journals. His main research directions include industrial data mining and 3-D visual perception.

Fengmao Lv (Member, IEEE) received the bachelor's and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2013 and 2018, respectively. He is currently an Associate Professor with Southwest Jiaotong University, Chengdu. His research focus includes transfer learning, domain adaptation, and deep learning.

Haonan Luo received the Ph.D. degree from Nanjing University of Science and Technology, China. He is currently working as an Assistant Professor at Southwest Jiaotong University, China. Previously, he was a Research Assistant at the Nanyang Technological University. He received the 2021 Best Paper Award for a paper published in Digital Image Computing: Techniques and Applications. He has several publications in top journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Information Fusion*, and ICCV. His research interests include deep learning, computer vision, embodied artificial intelligence, robotics, and reinforcement learning.

Gexiang Zhang (Senior Member, IEEE) received the Ph.D. degree from Southwest Jiaotong University, Chengdu, China, in 2005. He was a Visiting Professor with the Department of Computer Science, The University of Sheffield, U.K.; a Senior Visiting Professor at the Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Spain; and a Visiting Professor with the Department of Chemistry, New York University, USA. He is currently a Full Professor at the School of Automation, Chengdu University of Information Technology, Chengdu, China. He is the author/co-author of more than 200 publications and two monographs Research areas include artificial intelligence, natural computing, robotics, power systems, and their interactions. He is the President of International Membrane Computing Society (IMCS), a fellow and Fellow Assessor of the IET, the Editorial Board Member of Axioms and *International Journal of Parallel, Emergent and Distributed Systems*. He is the co-winner of Grigore Moisil Prize of the Romanian Academy in 2019. He is a (lead) Guest Editor/Co-Editor of more than ten volumes/proceedings.

Zhenghua Chen (Senior Member, IEEE) received the B.Eng. degree in mechatronics engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2011, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2017. He is currently a Scientist and the Lab Head of the Institute for Infocomm Research, and an Early Career Investigator with the Centre for Frontier AI Research, Agency for Science, Technology and Research (A*STAR), Singapore. His research interests include data-efficient and model-efficient learning with related applications in smart city and smart manufacturing. He is currently the Chair of IEEE Sensors Council Singapore Chapter.