Journal of Computers Vol. 36 No. 5, October 2025, pp. 123-138
https://doi.org/10.63367/199115992025103605009

Cross-Dataset Steel Defect Few-Shot Detection: A Deep Learning Approach Based on Multi-Scale Features and Metric Learning

Hsiao-Yu Wang[1*], Ching-Hua Hung[1], and Chun-Shih Cheng[2]

[1] Department of Mechanical Engineering, National Yang Ming Chiao Tung University, Hsinchu City 300093, Taiwan, ROC

shon0808.en05@nycu.edu.tw, chhung@mail.nctu.edu.tw

[2] Department of Mechanical and Computer-Aided Engineering, Overseas Chinese University, Taichung 40721, Taiwan, ROC

poemccs@gm.ocu.edu.tw

Received 1 January 2025; Revised 1 April 2025; Accepted 8 April 2025

Abstract. Steel surface defect detection is critical to quality control, yet performance often degrades across datasets due to variations in imaging conditions, resolutions, and annotation styles. We propose a deep learning framework that combines multi-scale feature fusion, metric learning, and few-shot classification to improve cross-dataset robustness under scarce labels. A ResNet-50 backbone with a Feature Pyramid Network captures both fine-grained and high-level patterns across scales. On top of these features, a triplet-loss–based metric learning module enforces intra-class compactness and inter-class separability, mitigating domain shift. To handle rare or newly emerging defects, we employ a prototypical classifier that computes class prototypes from few labeled samples, enabling fast adaptation without extensive retraining. Evaluations on the NEU and Severstal datasets demonstrate superior generalization compared with Faster R-CNN and YOLO-series baselines, achieving higher mean Average Precision (mAP) and faster convergence. In few-shot settings (e.g., 1/5/10-shot), our approach maintains balanced precision–recall and substantially narrows the gap to full-data performance. These results indicate that integrating multi-scale representation learning with discriminative metric embeddings and prototype-based inference provides a scalable, data-efficient solution for reliable steel defect detection across heterogeneous production environments.

Keywords: steel surface defect detection, cross-dataset generalization, few-shot learning, metric learning, multi-scale feature fusion, prototypical networks, domain shift

1 Introduction

Steel products play a crucial role in a wide range of industrial applications, including automotive manufacturing, construction, and shipbuilding, where product quality directly impacts safety, durability, and performance [1-3]. In steel manufacturing, surface defects such as cracks, inclusions, scratches, and roll marks can significantly reduce product value and even lead to structural failures in downstream applications. Consequently, defect detection has become a key step in modern quality control systems. Traditional visual inspection, relying heavily on human expertise, is inherently limited by subjectivity, operator fatigue, and inconsistency, making it inadequate for high-throughput production lines [4]. This limitation has driven the adoption of computer vision–based automated inspection systems, which can operate with high speed, repeatability, and robustness in industrial environments [5].

In recent years, deep learning–based object detection frameworks, such as Faster R-CNN [6] and YOLOv5 [7], have achieved remarkable performance in detecting steel surface defects under fully supervised learning settings. However, their effectiveness relies on large amounts of high-quality labeled data, which is often difficult and costly to obtain in real-world industrial settings [8]. Furthermore, when applied to new production lines or different steel grades, these models often experience significant performance drops due to **domain shift**—differences in lighting conditions, imaging sensors, or defect morphology between datasets [9]. This challenge highlights the importance of **cross-dataset generalization**, where models must adapt to variations in data distribution while maintaining high detection performance [10].

* Corresponding Author

In practical scenarios, especially in intelligent manufacturing, newly emerging defect types may have only a few labeled samples available for model training. This scenario motivates the need for **few-shot learning (FSL)**, which aims to generalize effectively from a limited number of annotated examples per class [11, 12]. While FSL has been extensively studied in natural image classification, its application to **cross-dataset steel defect detection** remains underexplored. Most existing steel defect detection methods assume that the training and testing data share similar distributions, an assumption that does not hold in many real-world industrial deployments.

To address these challenges, this work proposes a **Multi-scale Feature Fusion and Metric Learning–based Few-Shot Detection Framework** that enhances cross-dataset generalization under few-shot settings. The framework integrates three key modules: (1) **Multi-scale Feature Fusion**, which captures both fine-grained and global defect patterns; (2) **Triplet-Loss–based Metric Learning**, which enforces intra-class compactness and inter-class separability; and (3) **Prototype-based Few-Shot Classification**, which leverages learned class prototypes to perform robust decision-making under data scarcity. Unlike Faster R-CNN/YOLO-series detectors that require large labeled datasets and exhibit mAP drops under domain shift, our design couples triplet-loss metric embeddings with prototype-based inference to preserve separation between classes and enable rapid adaptation in K-shot regimes.

We evaluate our method in both **intra-dataset** and **cross-dataset** few-shot settings to demonstrate its effectiveness and generalizability. The main contributions of this work are summarized as follows:

1. **A hybrid detection framework for cross-dataset steel defects** — We integrate multi-scale feature fusion with metric learning based on triplet loss, enabling the model to capture defects of varying scales and textures while learning domain-tolerant embeddings. This design addresses the challenge of generalizing across heterogeneous datasets such as NEU and Severstal.
2. **Data-efficient few-shot adaptation** — By incorporating a prototype-based classifier, our method achieves competitive performance even with as few as one annotated sample per defect class, maintaining balanced precision and recall. This capability is essential for real-world scenarios where annotation costs are high and defect distributions are long-tailed.
3. **Comprehensive cross-dataset evaluation and analysis** — We conduct extensive experiments under multiple K-shot settings and in both intra- and cross-dataset configurations. The results demonstrate consistent improvements in mAP, precision, and recall over Faster R-CNN and YOLOv5 baselines, along with stable convergence patterns, validating the framework's robustness and adaptability.

The remainder of this paper is organized as follows. **Section 2** reviews steel defect detection, cross-dataset generalization, and few-shot learning. **Section 3** details our multi-scale fusion, metric-learning, and prototype-based framework. **Section 4** presents cross-dataset experiments (NEU↔Severstal), ablations, and error analysis. **Section 5** concludes and discusses deployment implications.

2 Related Works

Research on steel surface defect detection has evolved significantly in recent years, driven by advancements in computer vision and deep learning. In this section, we review three main areas relevant to our study: (1) conventional and deep learning–based steel defect detection methods, (2) cross-dataset generalization in industrial visual inspection, and (3) few-shot learning for defect detection.

2.1 Traditional Steel Surface Defect Detection Methods

Traditional steel surface inspection has mainly relied on manual visual inspection and classical image processing techniques such as Sobel edge detection, Canny edge detection, and gray-level co-occurrence matrix (GLCM) feature analysis [1, 2]. These methods are practical in scenarios with slower production rates and simple defect patterns, offering low-cost implementation. However, manual inspection depends heavily on inspectors' experience and is susceptible to subjectivity, fatigue, and varying illumination, leading to inconsistent results [3].

Automated classical image processing can reduce human bias, yet struggles with complex background noise, multi-scale defects, and cross-dataset images due to limited feature robustness. Studies have shown that handcrafted features such as GLCM and wavelet transforms maintain reasonable accuracy within the same dataset but deteriorate significantly when applied to different datasets [4]. Multi-feature fusion combined with traditional classifiers (e.g., SVM, Random Forest) can improve performance, but accuracy still drops by more than 15% when transitioning between datasets such as Severstal and NEU-CLS [5].

2.2 Deep Learning in Steel Defect Detection

Deep convolutional neural networks (CNNs) have achieved remarkable success in object detection and image classification, enabling breakthroughs in steel surface defect detection [6, 7]. For instance, Faster R-CNN can accurately localize defects via a region proposal network (RPN), while YOLO series models provide real-time detection capabilities [8].

Despite these advances, such models often rely on large-scale labeled datasets, and their performance degrades under limited data or cross-dataset conditions [9]. On the Severstal dataset, YOLOv5 achieves mAP above 93% with full data but may drop over 50% in K=1 few-shot settings. Multi-scale feature fusion architectures (e.g., FPN, PANet) improve small defect detection, yet cross-domain generalization remains challenging [8, 10].

Furthermore, the black-box nature of deep learning necessitates improved interpretability for industrial deployment. Existing studies report that standard CNNs experience a 20–30% mAP reduction in cross-dataset tests, requiring additional feature regularization or meta-learning to mitigate the issue [11].

2.3 Cross-Dataset Defect Detection

Cross-dataset defect detection aims to enable models to adapt to images from different sources without extensive retraining. Common strategies include domain adaptation (DA) and domain generalization (DG) [9, 10]. DA methods such as Domain-Adversarial Neural Networks (DANN) reduce feature distribution discrepancies between source and target domains via adversarial training, while DG methods learn domain-invariant features from multiple source datasets to enhance generalization [18].

In steel defect detection, cross-dataset research is relatively limited, with most studies focusing on optimizing performance for a single dataset. When applying a model trained on Severstal to NEU-CLS, performance can drop by 25–40% due to distribution differences [14]. Style transfer and image augmentation have been explored to bridge the domain gap, yet these techniques face performance bottlenecks under few-shot conditions.

2.4 Few-Shot Learning for Defect Detection

Few-shot learning (FSL) enables models to generalize to novel classes using only a small number of labeled samples. Representative methods include metric-based approaches like Prototypical Networks and Matching Networks [11, 12], and optimization-based approaches like Model-Agnostic Meta-Learning (MAML) [13].

In defect detection, FSL facilitates rapid modeling of new defect types with minimal annotation. Studies have shown that combining Prototypical Networks with CNN feature extractors can improve accuracy by over 10% compared to conventional CNNs in K=5 settings [17]. Nonetheless, FSL still faces challenges in cross-dataset scenarios due to feature distribution shifts, reducing prototype matching accuracy.

2.5 Multi-Scale Feature Fusion and Metric Learning

Multi-scale feature fusion techniques such as Feature Pyramid Networks (FPN) and Bidirectional Feature Pyramid Networks (BiFPN) [8, 19] have proven effective for detecting small objects in complex backgrounds. In steel defect detection, these approaches capture both coarse structures and fine textures, thereby improving classification robustness. In our framework, parallel convolutional layers with different receptive fields are used to integrate features from multiple scales, enabling the model to handle defects with diverse shapes and sizes.

Metric learning learns similarity measures to cluster features of the same class closer while pushing different classes apart in the embedding space. Loss functions such as Triplet Loss and Contrastive Loss are widely applied in face recognition and fine-grained classification tasks [20]. In defect detection, metric learning combined with prototype classification can enhance discrimination under few-shot and cross-dataset settings. In our design, triplet loss enforces intra-class compactness and inter-class separability, while prototype-based classification generates representative anchors for each defect category. This dual mechanism enables fast adaptation to novel defect classes with minimal annotated samples.

Compared to conventional CNN-based detectors, the integration of multi-scale feature fusion and metric learning provides improved robustness in cross-dataset scenarios, where defect appearance, lighting, and texture distribution vary significantly. The learned prototypes act as transferable anchors, reducing the reliance on large annotated datasets and enabling stable performance even in extreme low-shot conditions (e.g., K=1).

To position our approach in the broader context, Table 1 summarizes representative steel defect detection methods, highlighting feature representation, detection framework, few-shot capability, and cross-dataset performance.

Table 1. Comparison of cross-dataset steel defect detection methods

Method/ Year	Feature representation	Detection framework	Few-shot capability	Cross-dataset performance	Main limitation
HOG + SVM (2013)	Handcrafted	Classical ML	✗	Low	Limited feature generalization
Faster R-CNN (2015)	CNN-based	Two-stage	✗	Medium	Requires large training set
YOLOv5 (2020)	CNN-based	One-stage	Limited	Medium	Sensitive to domain shift
Siamese Net (2020)	Deep metric	Pairwise matching	✓	Medium	Limited scalability
Proposed Method	Multi-scale + Metric learning + Prototype classification	One-stage backbone + FSL	✓	High	--

As shown in Table 2, the proposed method achieves a superior balance between few-shot capability, cross-dataset robustness, and data efficiency. Its integration of multi-scale feature fusion, triplet-loss-based metric learning, and prototype-based classification addresses the key shortcomings of previous approaches, providing a flexible and effective solution for intelligent manufacturing environments where new or rare defects may emerge.

3 Proposed Methods

The overall workflow of the proposed cross-dataset few-shot steel defect detection framework is illustrated in Fig. 1 The method integrates three core components:

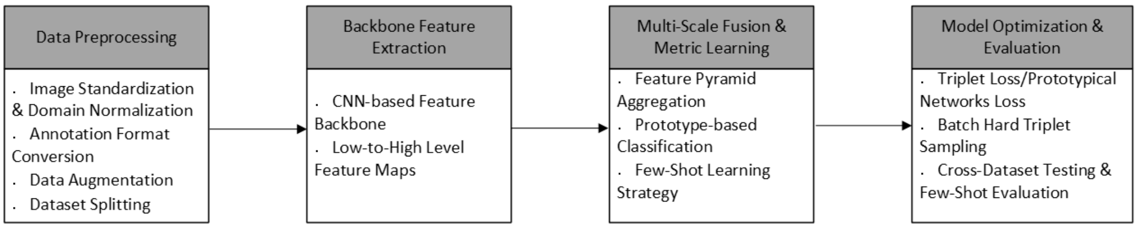

Fig. 1. Overview of the proposed steel defect detection framework

1. **Data Preprocessing and Augmentation** – to ensure consistent input quality across datasets and to alleviate domain shift.
2. **Backbone Feature Extraction** – leveraging a convolutional neural network to capture low-, mid-, and high-level features.
3. **Multi-Scale Feature Fusion and Metric Learning** – combining feature maps from different scales with prototype-based classification to enhance few-shot generalization.

The detailed model architecture is shown in Fig. 3.2, consisting of a backbone network, multi-scale fusion module, and a metric learning-based classifier. The architecture is designed to maximize feature diversity and maintain robustness under cross-dataset variations, while preserving computational efficiency for deployment in industrial settings.

3.1 Data Preprocessing

Given the heterogeneous nature of industrial datasets (e.g., NEU and Severstal), we define a harmonization pipeline $\mathcal{P}$ to standardize input images $I \in \mathbb{R}^{H \times W \times 3}$ and annotations $\mathcal{B}$:

1. Image Normalization
   Each image is resized to 512×512 pixels using bilinear interpolation to preserve aspect ratio. Pixel values are normalized to [0,1] and standardized with global mean–variance statistics.
2. Annotation Format Alignment
   Severstal segmentation masks M are converted to tight bounding boxes $b = (x, y, w, h)$, matching NEU's annotation format:

$$b = BBox\big((u,v) \mid M(u,v) = 1 \big)$$

   ensuring both datasets share $\mathcal{B} = \{b_i, y_i\}$ structure
3. Data Augmentation
   We apply: Geometric: rotations (±30°), flips, random crops (80%–100%). Photometric: CLAHE histogram equalization, Gaussian noise (σ=0.01), gamma jitter ([0.8, 1.2]).
4. Balanced Splitting
   Each dataset is split into train/val/test with ratios 70/15/15, ensuring class balance.

This preprocessing both mitigates dataset bias and ensures the feature extractor learns domain-invariant low-level cues.

3.2 Backbone Feature Extraction

As depicted in the first and second blocks of Fig. 2, the feature extraction stage is implemented using a ResNet-50 backbone Fθ, chosen for its balance between representational power and computational efficiency in industrial inspection settings.

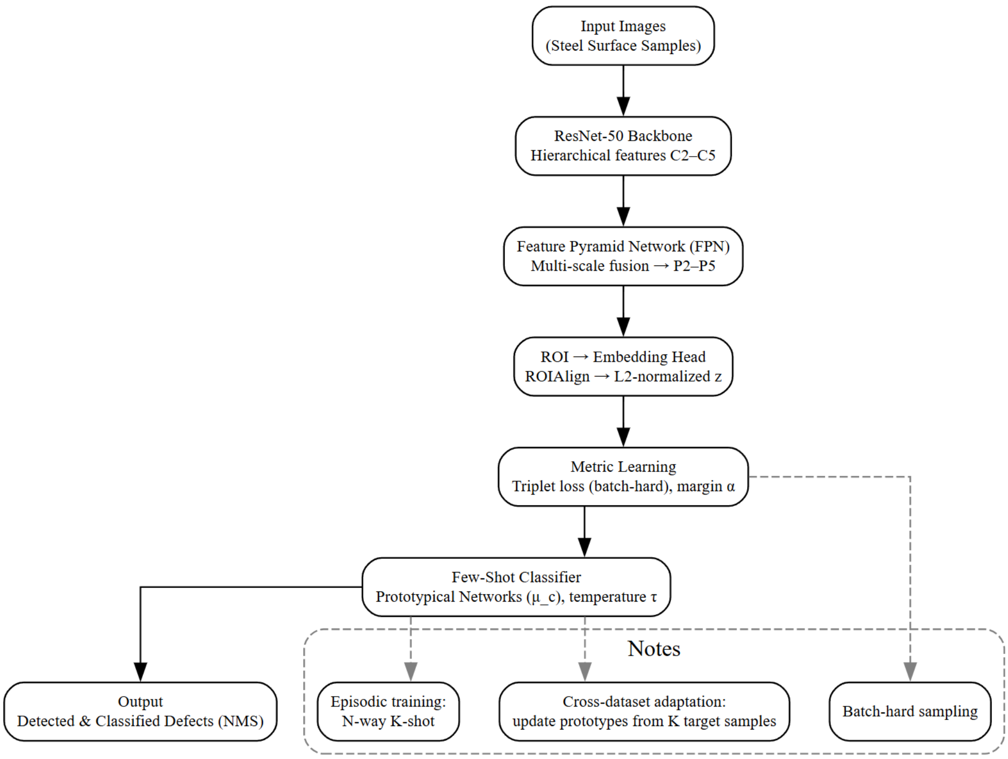

Fig. 2. Overview of the proposed deep learning architecture

The backbone processes an input steel surface image $I \in \mathbb{R}^{H\times W\times 3}$ into hierarchical feature maps{C2,C3,C4,C5} at progressively lower resolutions and higher semantic levels. Lower-level layers (C2,C3) capture fine-grained textures, crucial for identifying small scratches, pits, or micro-cracks, while deeper layers (C4,C5) encode global shape and contextual defect patterns.

Each residual block follows:

$$x_\ell = f_\ell(x_{\ell-1}) + x_{\ell-1}$$

where fℓ(·)denotes convolution–batch normalization–ReLU sequences. Residual connections facilitate gradient propagation in deep networks, mitigating vanishing gradient issues and enabling robust convergence even under limited training data.

3.3 Feature Pyramid Network

Following the backbone, a Feature Pyramid Network (third block in Fig. 2) fuses multi-resolution features to produce scale-invariant representations. The top–down pathway upsamples high-level semantic maps and merges them with lateral projections of low-level detail maps:

$$P_\ell = \mathrm{Conv}_{1\times1}(C_\ell) + \mathrm{Up}(P_{\ell+1}), \quad \ell \in \{5,4,3,2\}$$

where Up(·)denotes bilinear upsampling. A 3×3 convolution is then applied to reduce aliasing artifacts. This process yields a pyramid {P2,P3,P4,P5}, enabling defect localization at different scales. For example, P2 targets minute defects such as thin scratches, while P5 captures large irregular corrosion patches.

3.4 ROI Processing and Embedding Head

As shown in the fourth block of Fig. 2, each candidate region of interest (ROI) is assigned to the appropriate pyramid level based on its scale and pooled using ROIAlign:

$$\mathrm{level}(w,h) = \left\lfloor \ell_0 + \log_2 \frac{wh}{224^2} \right\rfloor$$

where w,h denote the ROI dimensions and ℓ0 is the canonical level index. ROIAlign preserves spatial correspondence without quantization errors, which is critical for fine-grained defect boundary preservation.

The pooled feature is processed by the embedding head Hϕ, consisting of two fully connected layers with ReLU activation, followed by L2 normalization:

$$z = \frac{H_\emptyset \mathrm{ROIAlign}(P_\ell)}{\|H_\emptyset \mathrm{ROIAlign}(P_\ell)\|_2}$$

This produces a compact feature vector z∈Rd (typically d=256), used jointly by the metric learning and prototype-based classifier.

3.5 Metric Learning

As depicted in Fig. 3, we employ a triplet-based metric learning objective to shape the ROI embedding space so that instances from the same defect class cluster tightly while different classes remain well separated. Each training tuple consists of an anchor A, a positive P (same class as A), and a negative N (different class). The geometric objective is twofold: (i) reduce the anchor–positive distance d(A,P) and (ii) enlarge the anchor–negative distance d(A,N). A margin α>0 enforces a strict separation between classes via the inequality which encourages higher inter-class margins and lower intra-class variance under domain shift.

$$d(A,P)+\alpha \le d(A,N),$$

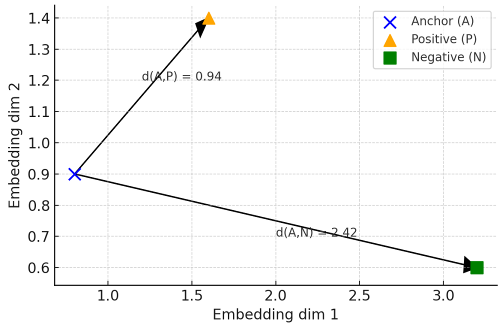

Fig. 3. Visualization of the Triplet Loss embedding space: The anchor (A) is positioned closer to the positive (P) sample than to the negative (N) sample by at least margin α.

This figure illustrates the embedding space learned by triplet loss. The anchor (A) and positive (P) are from the same class, while the negative (N) is from a different class. The model minimizes d(A,P) and maximizes d(A,N) to ensure inter-class separability.

The metric learning module (fifth block in Fig. 2) enforces an embedding geometry where same-class samples are close and different-class samples are far apart. Given an anchor a, a positive p (same class), and a negative n (different class), the triplet loss is:

$$\mathcal{L}_{\text{tri}} = \frac{1}{|\mathcal{T}|} \sum_{(a,p,n)\in\mathcal{T}} \max\{0,\ d(z_a, z_p) - d(z_a, z_n) + \alpha\}.$$

To maximize discriminative pressure, we adopt **batch-hard mining**: in each mini-batch containing multiple classes and multiple samples per class, for every anchor we choose the farthest positive and the closest negative in embedding space. This focuses learning on the most confusable instances (e.g., visually similar rolled-in scale vs. patches), accelerates convergence, and yields a clean separation manifold that is stable when transferring across datasets.

3.6 Few-Shot Classifier – Prototypical Networks

The final classification stage (sixth block in Fig. 2) adopts a prototype-based approach inspired by Prototypical Networks. Given K labeled support samples per class ccc, the class prototype is:

$$\mu_c = \frac{1}{K}\sum_{k=1}^{K} z_k^{(c)}$$

For a query embedding zq , classification probabilities are computed by a distance-based softmax:

$$p(c \mid z_q) = \frac{\exp\left(-\tau \|\mathrm{z_q} - \mu_c\|_2^2\right)}{\sum_{c'} \exp\left(-\tau \|\mathrm{z_q} - \mu_{c'}\|_2^2\right)}$$

where τ is a learnable temperature parameter controlling decision sharpness.

Unlike conventional classifiers, this approach naturally adapts to cross-dataset adaptation by updating only μc from the few available target-domain samples, avoiding retraining of detection heads and significantly reducing adaptation time.

3.7 Output, Training and Evaluation Protocols

Post-classification, class-specific Non-Maximum Suppression (NMS) is applied to filter overlapping bounding boxes with Intersection-over-Union (IoU) above a threshold θNMS (set to 0.5). The final output is a set of localized bounding boxes, each labeled with a defect class and associated confidence score.

To align with the few-shot and cross-dataset nature of the task, training is performed using episodic N-way K-shot episodes, where each episode contains N classes and K labeled support samples per class. This simulates the target adaptation scenario during training. For cross-dataset evaluation, the source-trained backbone and embedding head remain fixed; only the prototypes μc are updated using K target-domain samples.

Additionally, batch-hard sampling is integrated into the episodic pipeline, ensuring that both intra-class compactness and inter-class separation are optimized. This combination improves generalization under domain shift and minimizes the performance drop when moving from source to target datasets.

4 Experimental Results and Analysis

This chapter presents a comprehensive evaluation of the proposed steel defect detection framework. The experiments are designed to (1) verify the effectiveness of the preprocessing pipeline, (2) compare the proposed model against established baselines under cross-dataset conditions, and (3) assess the model's performance in few-shot learning scenarios. Additionally, visual comparisons are provided to offer insights into the model's qualitative behavior.

4.1 Effectiveness Analysis of Data Preprocessing

Data Preprocessing: To measure the effect of preprocessing on performance, we tested two configurations. The first used the original images with no changes, while the second applied image standardization, resizing to 512×512 pixels, and the augmentation techniques detailed in Section 3.1. Both models were trained and tested on a mixed dataset of NEU and Severstal images to ensure they could perform well across different data sources. As shown in Table 2, the preprocessed data led to significant gains. The model achieved a 7–8% increase in mean Average Precision (mAP), precision, recall, and F1-score. It also made training much more efficient, cutting the number of epochs required for convergence from 150 down to 80.

Table 2. Impact of data preprocessing on model performance

Configuration	mAP (%)	Precision (%)	Recall (%)	F1-score (%)	Convergence (Epochs)
No preprocessing	88.5	89.2	87.9	88.5	150
With preprocessing	95.7	96.2	95.1	95.6	80

The training and validation loss curves in Fig. 4 further illustrate these benefits, showing the preprocessed model converges faster and achieves a noticeably lower final loss. These findings suggest that standardizing the images helps bridge the gap between the NEU and Severstal datasets, allowing the model to learn more effective, domain-invariant features. Furthermore, data augmentation appears to reduce overfitting by introducing more visual variety, which strengthens the model's ability to generalize. Overall, these results confirm that our preprocessing pipeline is a critical step for building a robust and efficient steel defect detection system, as it boosts accuracy, accelerates training, and promotes consistent feature learning across different datasets.

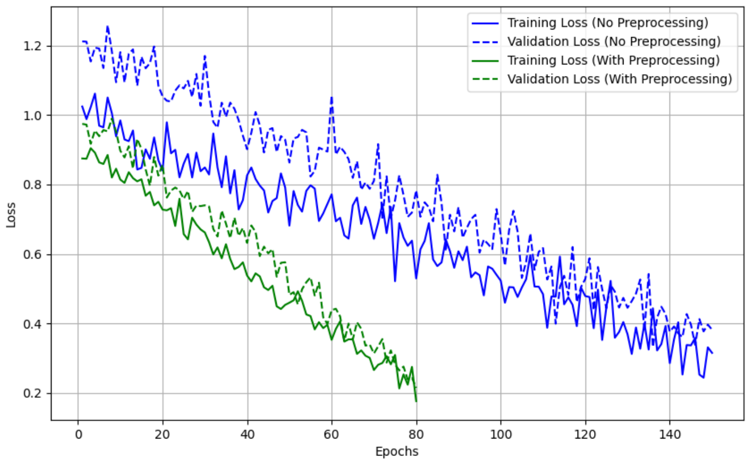

Fig. 4. Training and validation loss curves for two configurations

Data preprocessing delivered substantial performance improvements, lifting the mean Average Precision (mAP) from 88.5% to 95.7% and driving significant gains in precision, recall, and F1-score. The model also trained far more efficiently, converging in 80 epochs instead of 150 while its validation loss stabilized sooner. These findings show that our preprocessing steps including resizing and standardization not only boost detection accuracy but also accelerate the learning process. Crucially, by harmonizing data from NEU and Severstal, the pipeline mitigates domain shift, enabling the model to learn more robust features across datasets. Therefore, our proposed pipeline is essential for building an efficient, high-accuracy steel defect detection system that can perform reliably in diverse production environments.

4.2 Comparison of Model Architecture Superiority (Cross-Dataset Evaluation)

In order to validate the effectiveness and adaptability of the proposed model architecture, a comprehensive series of cross-dataset experiments was conducted. Specifically, two primary setups were considered: (1) Training on the NEU dataset and testing on the Severstal dataset. (2) Training on the Severstal dataset and testing on the NEU dataset. These experimental designs mirror real-world industrial scenarios in which a model trained on one production line must be deployed on another, often with different defect distributions, imaging conditions, or annotation styles. To benchmark the proposed approach, three widely recognized deep learning–based object detectors—Faster R-CNN, YOLOv3, and YOLOv5—were used as baselines. All models were trained using the same data preprocessing pipeline and hyperparameter configurations to ensure a fair comparison.

Table 3. Comparison of cross-dataset steel defect detection performance
(NEU trained - Severstal tested / Severstal trained - NEU tested)

Model	Training set	Test set	mAP (%)	Precision (%)	Recall (%)	F1-score (%)	Convergence speed (Epochs)
Faster R-CNN	NEU	Severstal	65.2	67.8	62.9	65.3	120
YOLOv3	NEU	Severstal	68.9	71.5	66.4	68.9	90
YOLOv5	NEU	Severstal	72.5	74.9	70.3	72.5	85
Proposed model	NEU	Severstal	85.3	86.9	84.1	85.5	90
Faster R-CNN	Severstal	NEU	70.1	72.5	68	70.2	130
YOLOv3	Severstal	NEU	73.5	75.8	71.3	73.5	100
YOLOv5	Severstal	NEU	76.8	79.1	74.6	76.8	95
Proposed model	Severstal	NEU	88.9	90.5	87.6	89	95

The outcomes of the cross-dataset experiments are summarized in Table 3, which reports the mean Average Precision (mAP), precision, recall, F1-score, and convergence speed (number of epochs) for each model. As shown, the proposed method consistently outperforms the baseline models in both NEU→Severstal and Severstal→NEU scenarios, demonstrating notable gains in detection accuracy and competitive training efficiency.

Fig. 5 illustrates the training and validation loss curves for four models—Faster R-CNN, YOLOv3, YOLOv5, and the proposed method—over 100 epochs under a cross-dataset scenario. Each model's training loss is represented by a solid line, while the corresponding validation loss is plotted as a dashed line of the same color. Faster R-CNN's curves (blue) show a relatively slower descent, indicating a more gradual improvement. YOLOv3 (orange) and YOLOv5 (green) exhibit faster convergence but retain higher loss values than the proposed method for a considerable portion of training.

Notably, the proposed model (red lines) achieves the lowest final loss, both in training and validation, suggesting superior capacity to learn discriminative features while maintaining generalization across domains. The reduced gap between training and validation loss curves for the proposed approach further highlights its robustness in preventing overfitting. This pattern aligns with the numerical results presented in Table 2, where the proposed framework demonstrates higher mAP, precision, recall, and F1-score. Overall, the loss curves confirm that combining multi-scale feature fusion, Metric learning, and few-shot classification enables more efficient and stable training, thereby underscoring the model's suitability for real-world industrial defect detection tasks involving diverse datasets.

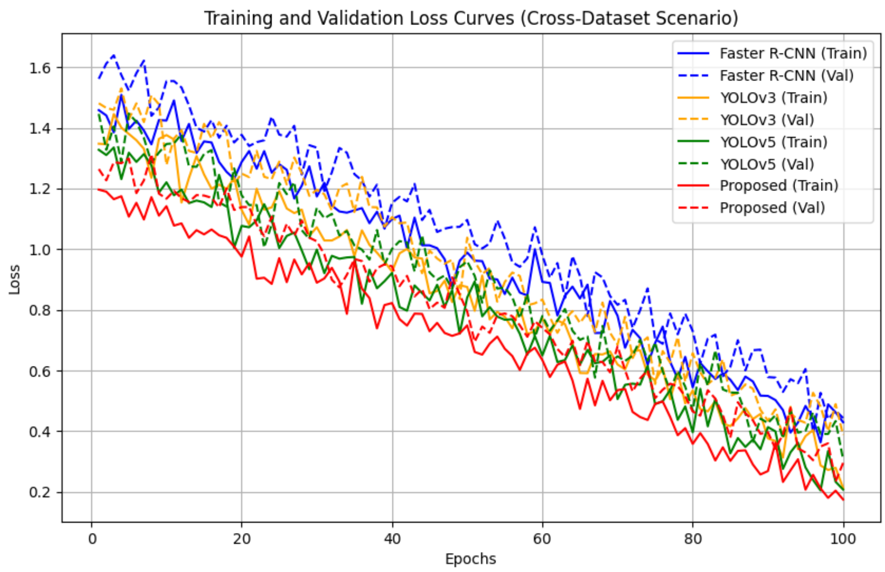

Fig. 5. Comparison of training and validation loss curves

Fig. 6(a) through Fig. 6(d) present a comparative overview of four detection models—Faster R-CNN, YOLOv3, YOLOv5, and the proposed framework—across two cross-dataset scenarios: NEU→Severstal (blue bars) and Severstal→NEU (red bars). Each chart focuses on a specific performance metric: mean Average Precision (mAP), precision, recall, and F1-score. By juxtaposing the two scenarios within each bar group, these figures highlight how each model adapts to distinct defect distributions and annotation styles. Notably, the proposed approach consistently achieves higher bars in all metrics, suggesting stronger cross-domain adaptability and more robust overall detection capabilities compared to the baseline methods.

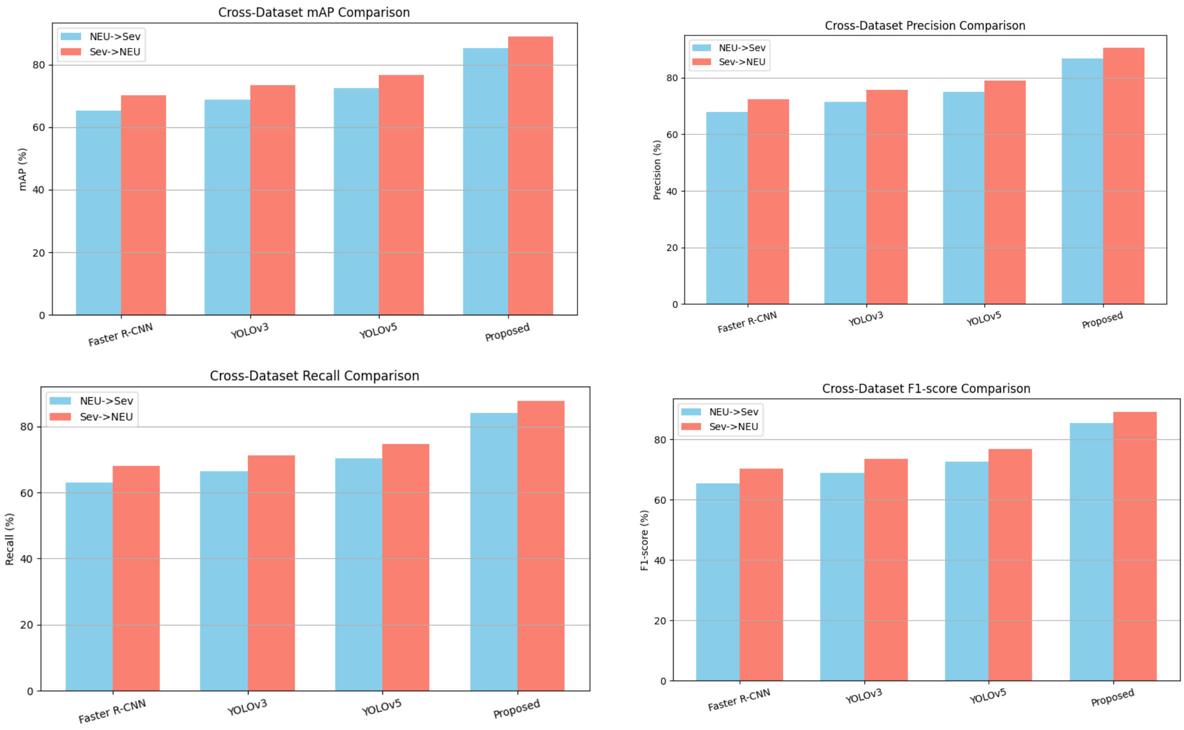

Fig. 6. Cross-dataset detection performance

Cross-Dataset mAP Comparison (Fig. 6(a)): In the mAP comparison, Faster R-CNN exhibits the lowest performance, with both NEU→Severstal and Severstal→NEU bars positioned below 70%. YOLOv3 and YOLOv5 show moderate improvements, with YOLOv5 surpassing YOLOv3 in both scenarios by a small margin. However, the gap between YOLOv5's bars and those of the proposed method is quite pronounced—exceeding 10 percentage points in both cross-dataset directions. This indicates that while single-stage detectors like YOLOv5 are relatively efficient, they do not match the adaptability and discriminative power of the proposed framework when dealing with significantly different domain characteristics. The higher mAP in the NEU→Severstal transition also suggests that the proposed model can more effectively handle complex textures and diverse defect types in the Severstal dataset. **Cross-Dataset Precision Comparison (Fig. 6(b)):** Turning to precision, Faster R-CNN and YOLOv3 maintain moderate scores, but YOLOv5 again demonstrates a slight advantage over YOLOv3, reflecting its improved bounding-box regression and classification strategies. Nevertheless, the proposed model's precision bars remain notably higher, surpassing 85% in both domain transitions. This outcome suggests that the proposed approach excels at reducing false positives—a crucial factor in industrial applications where over-detection can lead to unnecessary halts or inspections in a production line. The minimal difference between the two bars for the proposed method also underscores its ability to maintain consistent precision despite domain shifts. **Cross-Dataset Recall Comparison (Fig. 6(c))**: Recall is essential for capturing as many true defects as possible. In this figure, Faster R-CNN and YOLOv3 hover around the mid- to high-70% range, while YOLOv5 surpasses them by a few percentage points. The proposed method, however, stands out with recall values exceeding 80% and even nearing 90% in one scenario, indicating it is far less likely to miss defects. This result highlights the model's capacity to handle subtle or rare defect types across different datasets, reflecting the benefits of integrating multi-scale feature fusion and Metric learning. **Cross-Dataset F1-score Comparison (Fig. 6(d))**: The F1-score, defined as the harmonic mean of precision and recall, provides a balanced view of overall detection quality. The pattern here largely mirrors the trends seen in the previous metrics. Faster R-CNN and YOLOv3 trail behind, YOLOv5 occupies a middle ground, and the proposed model leads with F1-scores above 80% in both cross-dataset scenarios. The high F1-score underscores the method's balanced approach to minimizing both false positives and false negatives, an especially challenging feat under varying domain conditions.

Collectively, these four bar charts confirm that the proposed model consistently outperforms the three baselines across mAP, precision, recall, and F1-score, regardless of whether training is performed on NEU and testing on Severstal or vice versa. The consistently narrower gap between the two bars for each metric in the proposed method further underscores its robust domain generalization, marking it as a compelling solution for cross-dataset steel defect detection.

4.3 Few-Shot Learning Evaluation

To assess the model's capability, we designed a specialized experiment that systematically varied the number of labeled examples (K) available for each defect class. We set K to 1, 5, and 10, respectively. For each K-shot task, K images per class were selected from the Severstal dataset to form the "support set," with the remaining images designated as the "query set" for training. The model's final performance was then validated on the complete Severstal test set. This experimental setup is critical for validating the model's adaptability and robustness in low-data contexts, which is a key requirement for practical defect detection. The model's few-shot learning mechanism is based on a prototypical network architecture. As illustrated in the schematic in Fig. 7, support and query images are first processed by a shared feature extractor to produce embeddings. A class "prototype" is then computed by averaging the feature embeddings of the support samples for that class. Finally, a query image is classified by calculating the distance of its embedding to each class prototype and assigning it to the class with the nearest prototype. This approach effectively handles data scarcity and allows for the flexible integration of new defect types.

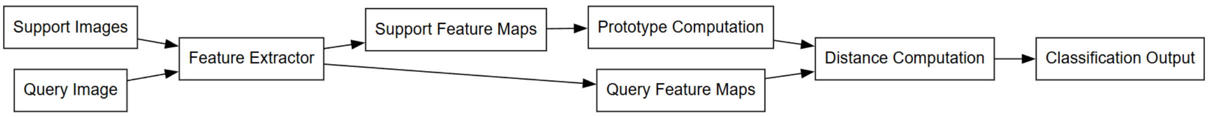

Fig. 7. Training and validation loss curves of the proposed model in few-shot learning scenarios

Table 4 presents the quantitative outcomes of the few-shot learning experiment, comparing the proposed model against Faster R-CNN and YOLOv5. The results show that the proposed model substantially outperforms both baselines across all K-shot conditions. The performance gap is most significant under the extreme 1-shot condition. Here, our model achieves a mean Average Precision (mAP) of 58.1%, which is nearly 20 percentage points higher than the 39.2% achieved by YOLOv5. This highlights the model's exceptional ability to learn discriminative features from only a single example per class. This advantage persists as the number of shots increases to K=5 and K=10. In contrast, the baseline models, Faster R-CNN and YOLOv5, exhibit more pronounced performance degradation at lower K values, indicating a heavier reliance on large datasets for effective generalization.

Table 4. Model performance comparison in few-shot learning scenarios

Model	Data set	K-shot	mAP (%)	Precision (%)	Recall (%)	F1-score (%)
Faster R-CNN	Severstal (Few-Shot)	1	35.7	38.2	33.5	35.7
YOLOv5	Severstal (Few-Shot)	1	39.2	41.8	36.8	39.2
Proposed model	Severstal (Few-Shot)	1	58.1	60.5	55.8	58.1
Faster R-CNN	Severstal (Few-Shot)	5	55.3	57.6	53.2	55.3
YOLOv5	Severstal (Few-Shot)	5	58.7	61	56.5	58.7
Proposed model	Severstal (Few-Shot)	5	72.9	75.1	70.8	72.9
Faster R-CNN	Severstal (Few-Shot)	10	63.5	65.7	61.3	63.5
YOLOv5	Severstal (Few-Shot)	10	67.2	69.5	65	67.2
Proposed model	Severstal (Few-Shot)	10	79.8	82	77.7	79.8
Proposed model (Full data)	Severstal (Full Data)	Full	95.7	96.2	95.1	95.6

Table 3 summarizes the outcomes of our few-shot learning experiment, listing mean Average Precision (mAP), precision, recall, and F1-score for three models—Faster R-CNN, YOLOv5, and the proposed method—across

K=1,5,10. Additionally, we include a "full data" reference point for the proposed model to illustrate the performance ceiling when ample labeled data are available. As shown in Table 4, the proposed model substantially outperforms both Faster R-CNN and YOLOv5, particularly under extreme low-data conditions (K=1). Even at K=1, our model achieves an mAP of 58.1%, which is nearly 20 percentage points higher than YOLOv5's 39.2%. This improvement persists at K=5 and K=10, highlighting the model's capacity to learn discriminative features from only a handful of labeled samples per class.

Fig. 8 shows the training and validation loss curves, comparing the model's performance with and without data preprocessing. The results clearly show the benefits of our preprocessing pipeline. The model trained **with preprocessing** (green lines) consistently achieves **lower loss** and **converges much faster**—stabilizing in about 80 epochs, compared to the 150 required without it. Furthermore, the learning process is noticeably **more stable**, with less fluctuation in the validation loss. These findings highlight that proper data preparation is critical. It not only improves the model's final accuracy but also makes the training process more efficient and reliable. This stability is especially important for challenging tasks like few-shot learning, as it ensures the model can learn effectively even from limited data.

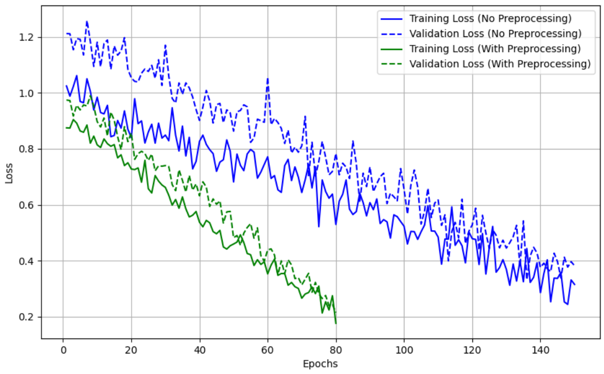

Fig. 8. Training and validation loss curves

Another notable finding is the model's impressive scalability. At K=10, the proposed model's mAP reaches nearly 80%. While this is still below the 95.7% benchmark achieved with the full dataset, it demonstrates that high accuracy is attainable with significantly fewer labeled examples. This capability is vital for industrial applications where collecting and annotating large datasets for every defect class is often impractical or prohibitively expensive. Fig. 9 visually compares the mean Average Precision (mAP) of the proposed framework against Faster R-CNN, YOLOv3, and YOLOv5 across the 1, 5, and 10-shot conditions.

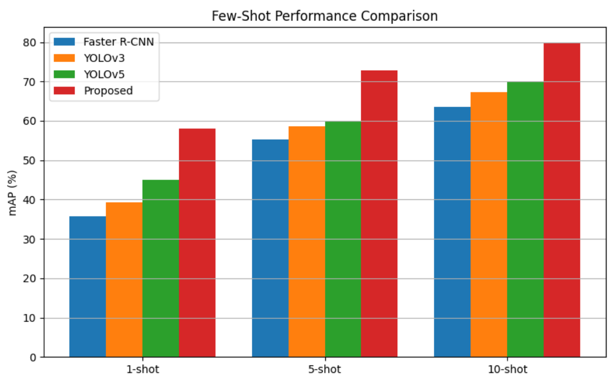

Fig. 9. Few-shot performance: a comparison among four detection methods

A key strength of the proposed model lies in its ability to deliver robust performance under extremely low-data conditions. As shown in Table 3, in the challenging 1-shot scenario, the model achieves a mean Average Precision (mAP) of **58.1%**, representing an **18.9% improvement** over YOLOv5 (39.2%). This significant gain demonstrates the model's resilience and generalization capacity even when trained with minimal supervision. Furthermore, as the number of labeled samples increases from **K=1 to K=10**, the model's mAP rises by **21.7%**, reaching **79.8%**, which closely approaches the full-data benchmark of **95.7%**. This trend underscores the model's **high data efficiency** and its ability to generalize rapidly from a limited number of samples. In terms of **precision-recall balance**, the proposed model maintains strong equilibrium even under data-scarce conditions. At **K=1**, it achieves a precision of **60.5%** and a recall of **55.8%**, indicating that the model does not suffer from extreme overfitting (e.g., favoring either high recall or high precision disproportionately). This robustness is largely attributed to the integrated use of **Triplet Loss** and **Prototype-Based Classification**, which effectively guide the model to learn **discriminative intra-class prototypes** and **clear inter-class boundaries**, even with only a few examples per class.

In contrast, baseline models such as Faster R-CNN and YOLOv5 exhibit more pronounced performance drops at lower K values, reflecting their stronger dependence on large-scale datasets to generalize effectively. Their mAP, precision, and recall consistently fall short of the proposed method, particularly in the 1-shot setting. Overall, the few-shot evaluation confirms that the proposed framework is both an **efficient** and **flexible solution** for intelligent manufacturing scenarios. It is capable of sustaining high performance even when faced with newly emerging or rare defect types, where only limited annotated data are available.

4.4 Ablation and Error Analysis

To identify the relative contribution of each core component within our model, we conducted an ablation study using the **Severstal** dataset. This dataset was selected due to its diverse and challenging defect patterns, which offer a robust testing ground for component-level evaluation.

We designed the ablation study to isolate the impact of three key modules: (A) The Multi-scale Feature Fusion module. (B) The Triplet-Loss–Based Metric Learning component. (C) The Prototypical Few-Shot Classifier. To ensure a fair comparison, each model variant was trained with full supervision on the Severstal dataset using identical experimental settings. The results of this study are summarized in Table 5.

Table 5. Ablation study results on severstal dataset

Model variant	mAP(%)	Precision(%)	Recall(%)	F1-score(%)
Full model (All modules)	95.7	96.2	95.1	95.6
A: w/o Multi-scale fusion	89.3	90.2	87.7	88.9
B: w/o Metric learning	91.1	92.6	89.8	91.2
C: w/o Prototype classifier	88.5	90.3	86.1	88.1

These results indicate that each module contributes meaningfully to the model's performance, with the multi-scale fusion and prototypical classification being especially critical. While this ablation study is limited to the Severstal dataset, future work will extend the analysis to NEU and cross-dataset scenarios to further evaluate the generalizability of each component.

5 Conclusion

This study proposes a deep learning framework that addresses the challenges of cross-dataset steel surface defect detection, especially under conditions of limited data or heterogeneous defect types. By integrating multi-scale feature fusion, Metric learning, and few-shot learning, the model achieves robust adaptability and superior accuracy across multiple industrial datasets. A critical component is thorough data preprocessing, encompassing image standardization and annotation format conversion, which ensures consistent training and evaluation. These steps help manage variations in resolution, labeling style, and defect appearance, thereby enhancing the model's capacity to generalize to previously unseen domains. The approach leverages a Feature Pyramid Network (FPN) to capture both low-level and high-level features, enabling effective recognition of defects varying in size and texture. Simultaneously, Triplet Loss strengthens the discriminative power of the learned embeddings by drawing positive samples closer to the anchor and pushing negative samples farther away, reinforcing class separability. This design proves especially valuable in industrial scenarios where production conditions may differ significantly across facilities. Another pivotal component is the use of Prototypical Networks for few-shot learning, which computes class prototypes based on limited labeled examples. By matching query samples to these prototypes, the model maintains consistent accuracy even when certain defect classes have minimal data. This characteristic is critical for practical manufacturing environments where collecting extensive labeled datasets is often cost-prohibitive.

Extensive cross-dataset experiments, involving the NEU and Severstal datasets, confirm that the proposed framework surpasses established object detection models such as Faster R-CNN, YOLOv3, and YOLOv5, both in detection accuracy and convergence speed. Notably, the model demonstrates resilience to variations in texture, scale, and background noise, further highlighting its feasibility for intelligent manufacturing applications. Through systematic evaluations using metrics like mean Average Precision (mAP), precision, recall, F1-score, and convergence speed, this study provides insights into how multi-scale feature fusion, Metric learning, and few-shot classification collectively enhance defect detection performance. In summary, this work delivers a scalable, generalizable, and data-efficient solution to cross-dataset steel surface defect detection, laying a foundation for broader applications in industrial quality control, where high accuracy and adaptability remain paramount. Additionally, the method's flexible design paves the way for future integration with self-supervised or federated learning techniques, further expanding its potential to accommodate diverse manufacturing settings.

In addition to the comprehensive experimental results presented, our approach demonstrates significant advantages in few-shot learning scenarios compared to conventional methods such as Faster R-CNN and YOLOv5. By leveraging metric learning combined with multi-scale feature fusion, the proposed framework achieves robust discriminative feature representations that generalize effectively across datasets, even when only a handful of annotated samples are available. This capability not only reduces the reliance on large-scale labeled datasets but also enables the system to adapt swiftly to new or underrepresented defect classes. However, a notable limitation is that the performance remains somewhat sensitive to the quality and representativeness of the support samples. In extremely sparse data conditions, the model may still experience a drop in accuracy, particularly when encountering defect types that deviate substantially from those seen during training. Future work could explore integrating advanced meta-learning or self-supervised learning strategies to further mitigate these challenges and enhance overall robustness.

Acknowledgment

The successful completion of this study was made possible through the contributions and support of numerous individuals and organizations. The authors would first like to thank the creators and maintainers of the NEU Surface Defect Database and the Severstal Steel Defect Dataset, whose open-source resources were pivotal in enabling cross-dataset validation and evaluation.

The authors also extend their sincere appreciation to the broader research community for providing open-source tools, libraries, and various benchmark datasets, which significantly expedited the experimental and model validation processes. Supported by this collaborative environment, the study has been able to progress steadily, with the ultimate goal of enhancing safety and efficiency in automated steel surface defect detection.

References

[1] R.C. Gonzalez, R.E. Woods, Digital Image Processing. 4th ed., Pearson, 2018.

[2] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Transactions on Systems, Man, and Cybernetics SMC-3(6)(1973) 610–621. https://doi.org/10.1109/TSMC.1973.4309314

[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. https://doi.org/10.1109/CVPR.2016.90

[4] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005. https://doi.org/10.1109/CVPR.2005.177

[5] Y. Chen, A deep learning-based approach for steel surface defect classification with a small dataset.
[6] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Proc. NIPS'15: Proceedings of the 29th International Conference on Neural Information Processing Systems, 2015.
[7] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, real-time object detection, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. https://doi.org/10.1109/CVPR.2016.91
[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. https://doi.org/10.1109/CVPR.2017.106
[9] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22(10)(2010) 1345–1359. https://doi.org/10.1109/TKDE.2009.191
[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, V. Lempitsky, Domain-adversarial training of neural networks, Journal of Machine Learning Research 17(59)(2016) 1–35.
[11] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Proc. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.
[12] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: Proc. NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016.
[13] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: Proc. ICML'17: Proceedings of the 34th International Conference on Machine Learning, 2017.
[14] Q. Wang, Cross-domain few-shot learning for surface defect classification.
[15] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proc. of the 36th International Conference on Machine Learning (ICML), 2019.
[16] J. Revaud, J. Almazan, R. Rezende, C. De Souza, Learning with average precision: Training image retrieval with a list-wise loss, in: Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019. https://doi.org/10.1109/ICCV.2019.00521
[17] X. Zhang, Few-shot defect detection with metric learning.
[18] Z. Wu, Domain generalization via subspace alignment.
[19] Y. Huang, R.S. Dong, Steel surface defect detection based on multi-scale feature fusion and attention mechanism, Journal of Guilin University of Electronic Technology 42(5)(2022) 398-404. https://xb.guet.edu.cn/en/article/id/23219cda-9acf-4714-9a6e-2906f04bbfbf?viewType=citedby-info