International Journal of Machine Learning and Cybernetics (2024) 15:4589–4601
https://doi.org/10.1007/s13042-024-02175-7

ORIGINAL ARTICLE

Yolo-sd: simulated feature fusion for few-shot industrial defect detection based on YOLOv8 and stable diffusion

Yihao Wen[1] · **Li Wang**[1]

Received: 4 September 2023 / Accepted: 8 April 2024 / Published online: 1 May 2024

Abstract
Defect detection from images, an important application in the development of the industrial internet, has been gaining increasing attention due to its close relationship with product quality in industrial production. However, two major challenges in defect detection persist: (1) Limited availability of datasets. Deep learning-based models typically require large-scale training sets to achieve satisfactory detection results. (2) Insufficient image quality and low detection accuracy. When many object detection methods are applied to industrial defect detection, they often exhibit poor performance in handling unclear boundaries, complex backgrounds, noise, and textures. In this study, we propose an advanced defect detection method based on YOLO and Stable Diffusion (YOLO-SD). For the few-shot dataset, a controllable generation module is designed, that integrates CLIP, LoRA, and ControlNet based on Stable Diffusion. Among them, CLIP text inversion can generate the most suitable prompt words from the defect dataset, providing prompt input for Stable Diffusion. LoRA can intervene and adjust the image style of Stable Diffusion by training on the defect dataset in a fine-tuning way. ControlNet obtains boundary and depth maps through HED and Midas. For the insufficient image quality and low detection accuracy, an improved YOLO model with an attention-based Fusion Simulated Feature module (FSF) is built that extracts defect features of the original images and generated images, which provides richer semantic information to improve the detection accuracy. At the same time, in order to make the model lightweight, we introduce a test optimization strategy to improve the model training process. Extensive experiments on the NEU-DET steel defect dataset show that the images generated by our method can expand the dataset to train the model and achieve a certain improvement in defect detection.

Keywords YOLO · NEU-DET · Stable Diffusion · Few-shot object detection

1 Introduction

Defect detection plays a vital role in industrial manufacturing [1, 2]. Surface defects generated during the production process, such as those found in steel, metal components, printed circuit boards, and ceramics, not only affect the appearance but also pose safety hazards. Therefore, surface defect detection is an essential component of industrial production [3]. In recent years, with the advancement of Internet of Things (IoT) technology, computer vision and deep learning methods have been widely applied in intelligent manufacturing, and deep learning-based automated surface defect detection is a prevailing trend [4].

Some deep learning-based methods have been published for detecting defects recently. For defect detection, two-stage methods, [5–7] based on Fast R-CNN and Faster R-CNN, have demonstrated high accuracy in detecting surface defects in various materials. However, their slow detection speed limits their applicability in industrial production scenarios. On the other hand, one-stage detectors, [8–15] based on YOLOv3, YOLOv4, and YOLOv5, offer faster detection speeds while maintaining satisfactory accuracy. These methods directly change defect detection to a regression task. They have been successfully applied to detect defects in steel, concrete, and other materials. However, in the presence of low-quality images, these approaches frequently exhibit suboptimal performance in handling unclear

✉ Li Wang
wangli@tyut.edu.cn

Yihao Wen
wenyihao238@163.com

[1] College of Computer Science and Technology (College of Data Science), Taiyuan University of Technology, 79 Yingze West Street, Taiyuan City 030024, Shanxi Province, China

boundaries, complex backgrounds, noise, and textures. Several enhancements have been proposed to improve the accuracy of defect detectors. These include incorporating attention mechanisms, such as the Convolutional Block Attention Module (CBAM) and the Efficient Channel Attention (ECA) mechanism, which enable the models to focus on more relevant features. Additionally, techniques like window shifting and transformer modules have been introduced to enhance feature extraction capabilities and expand the receptive field.

For few-shot object detection, detection networks designed based on meta-learning and transfer learning [16–19] utilize rich feature information from base categories and generalize to new categories. However, data quality greatly influences their performance, which requires careful adjustment based on specific data. Moreover, as data increases, they are susceptible to the problem of catastrophic disregard of original data. Efforts have been made in data augmentation [20], but these approaches only enhance the original images. When encountering unseen defects, they often struggle to identify them. Another approach involves manually modeling and generating a simulated image dataset [21] to address the issue of limited samples. Furthermore, zhai et al. performed synthetic data augmentation through a full-volume machine network to solve the steel fatigue crack identification problem [22], but the quality of the generated images still differs from real images, and the generated image's object positions are not fixed, requiring the re-annotation of generated images, which is time-consuming.

However, two major challenges in defect detection persist: (1) Limited availability of datasets in real-world applications. Conventional deep learning methods face challenges when dealing with new application scenarios, as they require extensive collections of new data for retraining or integration. This process is time-consuming, expensive, and carries the risk of overfitting. (2) Insufficient image quality and low detection accuracy. Due to factors such as variations in industrial settings, diverse types of defects, and differences in captured objects and environmental conditions during image acquisition, industrial defect detection often exhibits poor accuracy in real-world applications. Thus, the lack of datasets and insufficient accuracy have consistently presented significant obstacles in the field of defect detection.

To address the aforementioned challenges, we propose a defect detection model based on YOLOv8 and Stable Diffusion (YOLO-SD), which consists of a controllable generation module and a fusion simulated feature object detection module. First, we propose a controllable generative module based on Stable Diffusion, which integrates CLIP, LoRA, and ControlNet components. The CLIP model is used to extract prompt keywords from the defect dataset, which are then used as input for Stable Diffusion. LoRA fine-tunes Stable Diffusion using the defect dataset. Additionally, boundary and depth maps are obtained via HED and MiDaS and are provided to Stable Diffusion as conditional inputs through ControlNet. This ultimately generates results closer to real-world industrial scenarios and the generated defect positions are mostly aligned with the annotated positions in the original images. Then we propose a defect detection module based on YOLOv8. Our module involves synchronous extraction of features from both original images and images generated by the controllable generative module. Additionally, the YOLOv8 network is enhanced by a Fusion Simulated Feature module with attention (FSF) in the HEAD model, enabling the feature fusion of real features with simulated features generated at different reconstruction scales. Finally, through extensive experiments conducted on NEU-DET defect dataset, our YOLO-SD method achieves a mean average precision (mAP) of 82.3% on the NEUDET steel defect dataset, improved by 6.1% and 15.9% over the baseline YOLOv8 and YOLOv5 respectively. Furthermore, this paper achieves an mAP of 73.5% with the few-shot training set, improved by 3.7% over the baseline YOLOv8.

2 Related work

2.1 Defect detection

The objective of the defect detection task is to classify defects and determine their positions. Early researchers utilized traditional image processing methods to extract defect features, such as Histogram of Oriented Gradients (HOG) [23], Scale-Invariant Feature Transform (SIFT) [24], and Speeded Up Robust Features (SURF) [25]. Classification models like Support Vector Machines (SVM) were then employed to determine the defect categories [26]. The detection results of traditional methods heavily relied on the quality of handcrafted features, making them time-consuming and susceptible to various factors such as illumination, defect types, and environmental conditions.

Due to the rapid advancement of deep learning, Convolutional Neural Networks (CNNs) have gained significant traction in the field of object detection. Deep learning-based object detection algorithms can be broadly categorized into two paradigms: one-stage methods and two-stage methods. Within these paradigms, most defect detection networks are derived from object detection networks.

Two-stage methods have demonstrated high accuracy in surface defect detection. For instance, Jiao et al. [5] devised an end-to-end detection framework that combines anchor-free techniques with Fast R-CNN to identify surface defects in crops. Cha et al. [6] proposed an enhanced version of Faster R-CNN, modifying the region proposal network of ZFNet to detect concrete cracks and layered defects in steel. Hou et al. [7] introduced a Cascade Mask R-CNN with a transfer learning strategy for cable defect detection.

However, the slow detection speed of these two-stage methods poses a significant limitation in industrial production scenarios.

In contrast, one-stage detectors offer improved detection speed. Li et al. [8] treats object detection as a regression task, directly predicting anchor points that may contain objects on the feature map. Xing et al. [9] presented an enhanced YOLOv3 model that incorporates an additional prediction layer and residual connections between layers for surface defect detection on steel rails. Li et al. [10] proposed an improved YOLOv4 algorithm, integrating Convolutional Block Attention Module (CBAM) and Receptive Field Block (RFB) for detecting surface defects on steel strips. Ma et al. [11] introduced a lightweight detector based on YOLOv4, incorporating a dual-channel attention module for detecting surface defects on aluminum strips. By leveraging attention mechanisms, the entire network focuses on capturing more detailed information, resulting in improved detection accuracy. Ying et al. [12] constructed a model based on YOLOv5 embedding with the Efficient Channel Attention (ECA) mechanism, effectively detecting surface defects in steel wire braided hoses. This method accounts for the detection of ultra-small targets by replacing the smallest-scale prediction layer with a larger-scale prediction layer, enabling higher accuracy in ultra-small object detection. Li et al. [13] designed a defect detection model based on YOLOv5, employing light calibration and patching for insulation defect detection. Gao et al. [14] devised a novel window shifting scheme for the Swin Transformer, enhancing the model's feature extraction capability. Guo et al. [15] combined YOLOv5 and Transformer modules to expand the receptive field, enabling accurate prediction of surface defects.

2.2 Few shot object detection

In recent years, significant breakthroughs have been made in the field of few-shot learning. Several models have been proposed to address the task of few-shot classification [27–30]. Li et al. [16] followed the meta-learning strategy, multiple meta-training tasks are constructed on base categories to learn class-agnostic feature extraction, which is then applied to new categories. Wang et al. [17] proposed a transfer learning-based method and introduced a Multi-scale Positive Sample Refinement (MPSR) approach to address the scale variation problem in few-shot object detection.

The progress in few-shot object detection on natural images has facilitated research on industrial images. Cheng et al. [18] adopted a two-stage training scheme and designed metric-based detection to compute distances between representations of each class and perform classification tasks. Wang et al. [19] presented a few-shot defect detection method based on the publicly available NEU-DET [31] dataset. Two domain generalization methods were developed to enhance the appearance of new features. While comparable results were achieved, the model performance was only evaluated on new data, which could lead to catastrophic disregard of the original data. Many researchers have addressed the few-shot problem through data augmentation. For example, Zoph et al. [20] proposed a data augmentation-based method that achieved good results by oversampling and duplicating small objects. However, the model still relied on semantic training from existing data and exhibited weak generalization to new images. Siu et al. [21] simulated synthetic images of sewer pipes using 3D modeling in a virtual environment to address sewer pipe defect detection. Zhai et al. [22] proposed the use of randomly textured mappings onto 3D models to synthesize data for enhancing the steel bridge crack dataset. These methods effectively alleviate the burden of data collection and annotation in defect detection, but they rely solely on manually generated 3D models, which may differ from real data.

In summary, although deep learning-based detection methods have made significant progress in various industrial applications, they still require large amounts of training data. When facing few-shot problems, defect detection models struggle to generalize well across all categories. Existing methods for addressing few-shot problems, such as meta-learning and transfer learning, have achieved certain results. However, some of these methods require continuous incremental data collection, which can lead to catastrophic disregard of the original data. Synthetic data augmentation has partially addressed the few-shot problem, but existing methods often generate synthetic data either through manual modeling, resulting in significant differences from the original images, or by directly enhancing the original images, which typically yields subpar results when encountering unseen data. Recently, the emergence of diffusion-based image generation models [35] has led to unprecedented image quality, even in generating results at larger reconstruction scales that are indistinguishable from real images. In the field of industrial defect detection, there is currently little work that employs advanced generative networks for synthetic data augmentation. This paper proposes a synthetic data augmentation approach based on the Stable Diffusion-based controllable generative network. Furthermore, the latest version of the YOLO model, YOLOv8, is improved by incorporating a fusion model to improve the detection accuracy.

3 Method

In this research, a model named YOLO-SD is proposed, which encompasses a controllable image generation module based on Stable Diffusion 1.5 and an improved version of the

YOLOv8 object detection Module. The controllable generative module is utilized to generate 4 sets of simulated images with different reconstruction levels. In the object detection module, BACKBONE and HEAD1 modules employ the latest YOLOv8 backbone structure to synchronously extract features from the original image and corresponding simulated images. In the HEAD2 modules, a Fusion Simulated Feature module (FSF) is designed for the fusion of original features and simulated features. The entire YOLO-SD model framework is shown in Fig. 1.

In our model, Stable Diffusion, an image model from stability.ai, is introduced that employs the Latent Diffusion Model (LDM) [35] with forward and reverse diffusion. In the forward phase, the noise gradually disrupts the image into random noise. In the reverse phase, Markov chains remove predictive noise from Gaussian noise, enhancing stability. It excels at diverse image generation and defect replication, valuable for real-world industrial use. Stable Diffusion 1.5 introduces time-consistent training, improving quality and stability. In order to solve the problem of the few samples and generate higher quality images, we use Stable Diffusion as the basic model in the controllable generation module.

As the representative method of single-stage detectors, YOLOv5 from Ultralytics hub has always maintained the characteristics of high accuracy and efficiency. It has been applied to numerous industrial scenarios. Recently, Ultralytics hub released the next major updated version of YOLOv5, YOLOv8, introducing new features and improvements to further enhance performance and flexibility. Specific innovations include a new backbone network, a new Anchor-Free detection head, and a new loss function. Therefore, to address the problem of defect detection, we use YOLOv8 as the base model in the object detection module.

In summary, a controllable and high-quality image generation module is introduced, utilizing Stable Diffusion to align generated defect images with specific dataset styles and contours. Through a carefully designed workflow and parameter configuration, this system achieves precise and high-quality defect image generation. Generated images no longer require manual labeling, significantly reducing the need for manual annotation efforts.

3.1 Controllable generative module based on stable diffusion

To generate controllable and high-quality defect images, this paper proposes a scheme for image generation based on Stable Diffusion. It can generate images with a similar style to the picture dataset and control the generation process of defect images precisely according to the defect contours. The module framework is shown in Fig. 2.

The controllable generative module based on Stable Diffusion consists of four main components: CLIP, LoRA, ControlNet. The specific workflow is as follows:

First, this module employs the textual inversion feature of the Contrastive Language-Image Pre-training (CLIP) model [36] to generate prompt keywords corresponding to each category of the original images.

Second, the Stable Diffusion 1.5 generation model is fine-tuned using Low-rank adaptation (LoRA) [37] on a defect dataset, enabling it to learn the style, texture, and other characteristics of steel defects.

Third, by leveraging the HED (a Holistically-nested edge detection method) [38] and Midas (a depth map estimation method) [39] models, approximate contours and depth maps of the images are obtained. The ControlNet [40], utilizing the original input features, contour features, and depth map features, assists the generative network in achieving controllable generation. As a result, the generated defect positions in the images align closely with the original images, allowing for the reuse of existing annotations and alleviating the substantial annotation effort involved in generating images.

Finally, during the actual generation process, we design 4 groups of parameters representing different levels of reconstruction to generate images. Parameters with lower reconstruction levels generated images closer to the original,

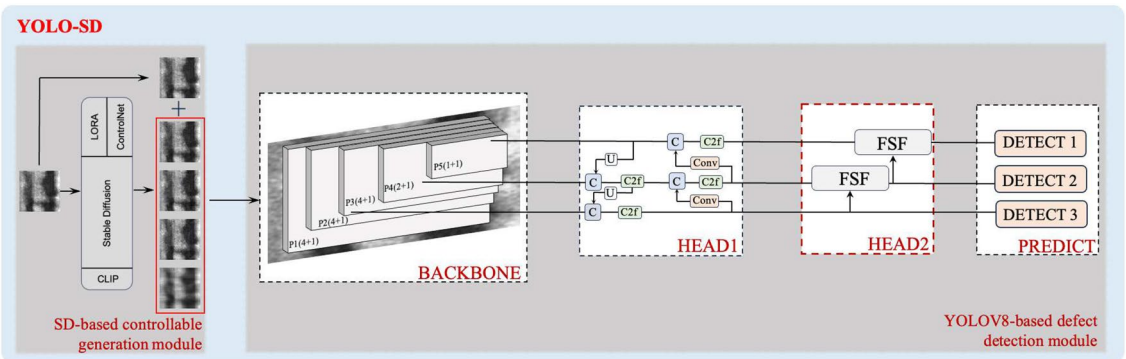

Fig. 1 The architecture of YOLO-SD. It consists of an SD-based controllable generation module and a YOLOV8-based defect detection module

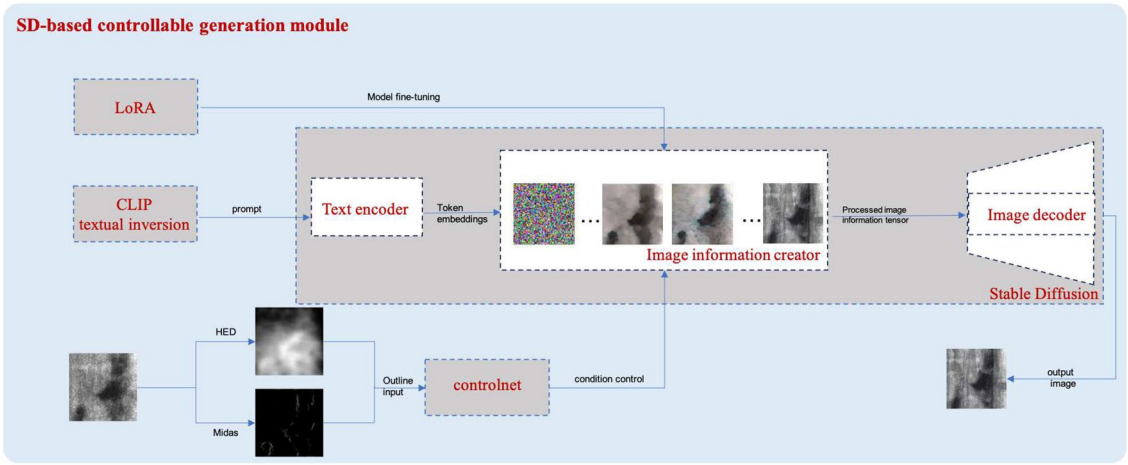

Fig. 2 The architecture of SD-based controllable generation Module. CLIP, LoRA, and ControlNet are integrated based on Stable Diffusion

while parameters with higher reconstruction levels introduced more variations. The specific parameter configurations are presented in Sect. 4.4.

3.2 Enhanced object detection module

To enhance the semantic richness and alleviate global class confusion in industrial defect images with large-scale variance, complex scenes, and limited samples, it is necessary to collect features from a vast training set and establish correlations to capture abundant semantic relationships.

Unlike conventional approaches, this paper does not directly augment the training set with simulated images. Instead, considering that excessive reliance on simulated image features may lead the network to become insensitive to original images during the learning process, it uses simulated images to fuse simulated features. The model architecture diagram is depicted on the right side of Fig. 1.

3.2.1 Model framework

The overall diagram of the YOLO-based object detection module consists of 4 main parts: BACKBONE, HEAD1, HEAD2, and PREDICT.

Wherein, the BACKBONE and HEAD1 modules employed YOLOv8 backbone structure to synchronously extract features from the original image and corresponding simulated images, then incorporate the pyramid structure of PANet [41] for the multi-scale fusion of images in HEAD1. PANet is a bidirectional fusion network that enhances feature representation through a bottom-up pathway. It leverages the precise localization information from lower levels to bolster the entire feature hierarchy, thus shortening the information path between the lowermost and uppermost layers.

In the HEAD2 modules, an FSF module is designed as illustrated in Fig. 3. In FSF module, the multiple feature maps that are interconnected undergo channel adjustment by applying a 1x1 convolutional layer. The adjusted feature map is then divided into three paths. The first path enters a series of bottleneck blocks for further non-linear transformations and feature fusion. The second path directly undergoes another 1x1 convolutional layer for channel adjustment. The third path undergoes channel attention and spatial attention like CBAM [42]. The feature maps processed by the CSP bottleneck blocks, attention feature maps, and channel-adjusted feature maps are concatenated and further processed by a 1x1 convolutional layer to obtain the final output feature map. The formula for the third path is as follows:

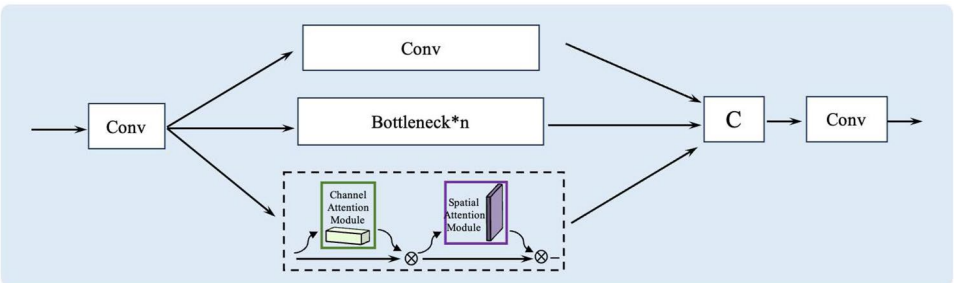

Fig. 3 The details of the FSF module, which adds channel attention and spatial attention

$$\mathbf{F_m} = \mathbf{A_c}(\mathbf{F_i}) \otimes \mathbf{F_i}$$
$$\mathbf{F_o} = \mathbf{A_p}(\mathbf{F_m}) \otimes \mathbf{F_m} \quad (1)$$

Let $\mathbf{F_i}$ represent the input feature map, $\mathbf{F_o}$ represent the output feature map, $\mathbf{F_m}$ represent the intermediate feature map, $\mathbf{A_c}$ denote channel attention, and $\mathbf{A_p}$ denote spatial attention. The $\mathbf{F_o}$ is obtained from $\mathbf{F_i}$ through $\mathbf{A_c}$ and $\mathbf{A_p}$.

For the $\mathbf{A_c}$, the spatial information of a feature map is aggregated by employing both average pooling (AvgPool) and max-pooling (MaxPool) operations. Subsequently, both descriptors are passed through a shared network to generate the channel attention map, denoted as $\mathbf{A_c} \in \mathbb{R}^{C\times1\times1}$. This shared network comprises a multi-layer perceptron (*MLP*) with a single hidden layer. The specific formula for $\mathbf{A_c}$ is as follows:

$$\mathbf{A_c}(\mathbf{F}) = \sigma(MLP(\text{AvgPool}(\mathbf{F})) + MLP(\text{MaxPool}(\mathbf{F}))) \quad (2)$$

where σ denotes the *sigmoid* function.

For the $\mathbf{A_p}$, the process begins with the application of AvgPool and MaxPool along the channel axis. These operations are performed to obtain an efficient feature descriptor that captures essential information. By conducting pooling along the channel axis, the aim is to emphasize informative regions within the feature map. The resulting concatenated feature descriptor is then processed through a convolutional layer, which generates a spatial attention map $\mathbf{A_p}(\mathbf{F}) \in \mathbb{R}^{H\times W}$. This attention map encodes the regions that require emphasis or suppression, indicating where to focus within the feature map. the specific formula for $\mathbf{A_p}$ is as follows:

$$\mathbf{A_p}(\mathbf{F}) = \sigma\left(f^{7\times7}([\text{AvgPool}(\mathbf{F});\text{MaxPool}(\mathbf{F})])\right) \quad (3)$$

where σ denotes the sigmoid function and $f^{7\times7}$ represents a convolution operation with the filter size of 7×7.

In the PREDICT module, [5, 3, 2] feature maps are fed into the detection heads corresponding to their respective scales for object detection. Considering the varying importance of information contributed by the feature maps from the original image and the feature maps combined from the original and simulated images, the original loss function of YOLOv8 is employed, and different weights are assigned to them during the parameter update of the SDG optimizer. For example, in the first scale of the *i* optimization iteration, a total of five input feature maps were considered. Let Θ represent the parameters from the previous iteration, and Θ' denote the parameters after applying the stochastic gradient descent (SGD) algorithm on a randomly selected sample. The weights W_k are assigned to each feature map to modulate their contribution to the parameter updates. Here, y^i denotes the actual target value or label, and $h_\Theta(x^{ik})$ represents the model's predicted value for the input sample x^{ik}. The parameter update formula is as follows:

$$\Theta' = \Theta + \sum_{k}^{5} W_k\left(y^i - h_\Theta\left(x^{ik}\right)\right)x^{ik} \quad (4)$$

The operational workflow of the entire model is as follows:

Initially, an image of a training set is processed through the controllable generative module to generate 4 reconstructed images of varying scales. Subsequently, the five images undergo sequential extraction of multi-scale features using the BACKBONE and HEAD1 modules. Notably, the features from the P3, P4, and P5 layers are involved in the subsequent fusion process within the HEAD2 module.

For the P3 layer, the corresponding outputs of the five HEAD1 modules are directly transmitted to the Detect3 module. Regarding the P4 layer, the HEAD1 output of the original image is directly transmitted to the DETECT2 module. Following this, the fusion of the HEAD1 outputs from the second and third images with the HEAD1 output of the first image, facilitated by the FSF module, is transmitted to the DETECT2 module. Similarly, the fusion of the HEAD1 outputs from the third and fourth images with the HEAD1 output of the first image is transmitted to the DETECT2 module. Concerning the P5 layer, the HEAD1 output of the original image is directly transmitted to the DETECT1 module. Subsequently, the fusion of the HEAD1 outputs from the remaining four images with the HEAD1 output of the first image, facilitated by the FSF module, is transmitted to the DETECT1 module.

Ultimately, two detections are performed in Detect1 (1 and 1+2+3+4+5), three detections in Detect2 (1, 1+2+3, 1+4+5), and five detections in Detect3 (1, 2, 3, 4, 5).

Through this design, the simulated images with different reconstruction levels are fully utilized, and the contributions of the original image and simulated images are balanced in the detection task. This approach helps improve detection accuracy and robustness and effectively leverages the relevant information between the original image and simulated images.

3.2.2 Lightweight testing strategy

On the other hand, in industrial inspection processes, ensuring both detection accuracy and speed is crucial. However, YOLO-SD requires the generation of images through a controllable generative network as a preliminary step. However, this process leads to prolonged preparation time, which hinders the real-time detection requirements. To address this issue, this paper separates the training and validation stages of the network.

During the training stage, the improved network structure is utilized to process the original and simulated images separately through the backbone and learn the parameters of the detection heads. This approach fully leverages the

features from the simulated images to enhance the network's accuracy. Through this strategy, a network model suitable for industrial inspection demands can be trained.

During the validation phase, without generating simulated images, the features outputted by the backbone network are duplicated into five copies and fed into the subsequent network. Furthermore, the detection head parameters obtained during the training phase are utilized. This ensures that the network maintains an acceptable detection speed (74 FPS in RTX3090)in practical applications while continuing to benefit from the advantages of simulated feature fusion.

The strategy of separating the training and validation stages aims to balance the requirements of detection accuracy and real-time performance. By fully utilizing the features generated from simulated images during network training and maintaining a small model parameter size and the original structure during the validation stage, this paper achieves accurate and efficient object detection in industrial inspection processes.

4 Experiments

4.1 Dataset

The NEU-DET dataset used in this paper is a collection of steel surface defect images curated by Northeastern University (NEU-DET) [32]. It includes six types of typical surface defects found in hot-rolled steel strips: cracks, inclusions, patches, pitted surfaces, rolled-in scales, and scratches. The dataset consists of 1800 grayscale images, with 300 samples for each defect type. Each sample image may contain multiple defects. The dataset is divided into training and testing sets with an eight-to-two ratio. The training set consists of 240 samples per class, while the testing set has 60 samples per class. Additionally, a few-shot partition scheme is designed, where 90 samples per class are randomly selected for training, and 60 samples per class are allocated for testing.

4.2 Evaluation metrics

To evaluate the performance of the YOLO-SD method, mean average precision (mAP) is used as the primary measurement metric. mAP represents the average precision across different classes. When evaluating an algorithm, an intersection over the union (IoU) threshold between Ground truth (A) and Prediction (B) is set to determine whether a detection is correct or not. If the IoU of a detected bounding box exceeds the threshold, it is considered a valid detection. mAP calculates the average precision at an IoU threshold over 50%. It evaluates the trade-off between recall and precision in the predicted bounding boxes for different object classes. The recall represents the matching between detected positive samples and true positive samples, and precision represents the accuracy of the detected positive samples, serving as the main ranking metric. IoU is calculated using the following formula:

$$IoU = \frac{A \cap B}{A \cup B} \tag{5}$$

In addition to accuracy, another crucial evaluation criterion for object detection tasks is speed, which plays a vital role in real-time applications. To achieve a more universal assessment of model speed, the use of frames per second (FPS) as the standard metric is avoided, as it varies across different operating environments. Instead, floating-point operations (FLOPs) serve as an indicator of computational workload, representing the number of floating-point operations required during model execution. FLOPs can be employed to measure the complexity of algorithms or models. In this study, FLOPs are adopted as the benchmark for assessing speed.

4.3 Experiment settings

In the SD-based controllable generative module, the following parameters are shared: Model: StableDiffusion v1.5, Sampler: DPM++ 2 M Karras, Seed: 865985793, CFGscale: 7.0. The generated image results in this paper are obtained using a CFG ratio of 7.0. The DPM++2 M Karras sampler, which is the default choice, is employed. Controlled generation experiments are conducted using the following prompt as textual input of Stable Diffusion:

CLIP textual inversion is used to generate keywords from the defect dataset. The keywords are sorted based on their occurrence frequency, with the top 40% selected as prompts. These prompts are then combined with defect categories and LoRA-style weights as follows: " grayscale, greyscale, hot-rolled steel strip, monochrome, no humans, surface defects, texture, grayscale, rolled-in scale, loRA:neudet1-v1:1". Subsequently, a range of generative parameter configurations was devised to encompass five distinct generation schemes with varying reconstruction scales, as shown in Table 1.

The generated images are illustrated in Fig. 4. Images with lower levels of reconstruction exhibit change primarily in texture, while as the degree of reconstruction increases, the images gradually undergo shape variations as well.

In the YOLO-based object detection model, SGD (Stochastic Gradient Descent) is utilized as the optimizer with a default weight decay of 0.0005 and a momentum of 0.937. Regarding learning rate scheduling, a warm-up strategy is employed to initialize the learning rate. During the warm-up phase, the learning rate is updated using linear interpolation at each iteration. Once the warm-up phase is completed, the switch is made to the cosine annealing algorithm

Table 1 Generation schemes incorporating multiple reconstruction scales

Parameters	Generate 1	Generate 2	Generate 3	Generate 4
LoRA Weight	1	1	1	0.9
Steps	40	60	80	120
Denoising strength	0.2	0.3	0.4	0.6
ControlNet-HED Weight	1	0.7	0.4	0.1
ControlNet-Midas Weight	1	0.7	0.4	0.1

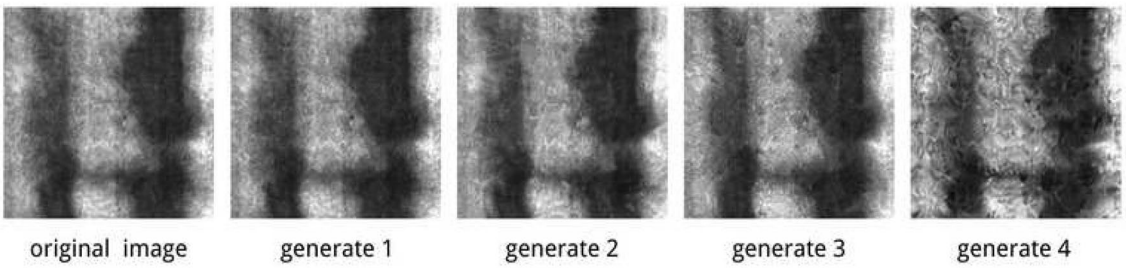

Fig. 4 Examples of generated defect images

Table 2 The mAP comparison results on NEU-DET

	faster-RCNN	Retina-Net	YOLOv5	YOLOv8	Ours
Crazing	37.6%	45.9%	47%	**50.5%**	**54.4%**
Inclusion	80.2%	**84.2%**	78.2%	79.9%	**84.6%**
Patches	85.3%	91.1%	**92.2%**	91.9%	**93.6%**
Pitted surface	**81.5%**	74.7%	77.1%	81%	**86%**
Rolled-in scale	54%	43.5%	44.9%	**68.7%**	**77.5%**
Scratches	89.2%	81.6%	71%	**95.1%**	**95.6%**
mAP	71.3%	70.2%	65.7%	**77.9%**	**82.3%**
FLOPs(G)	180	**53.4**	**49**	78.9	84.4

Table 3 The mAP Comparison results on few-shot NEU-DET

	YOLOv5	YOLOv8	Ours
Crazing	42.7%	35.7%	**49.2%**
Inclusion	70%	73.2%	**75.3%**
Patches	91%	90.3%	**92.2%**
Pitted Surface	66.7%	74.8%	**80.4%**
Rolled-in Scale	52.2%	53%	**59.5%**
Scratches	62.7%	**89.7%**	82.7%
mAP	64.2%	69.4%	**73.1%**

to update the learning rate. In the initial training phase of the model, warm-up training is conducted for a duration of three epochs, with the momentum of the SGD optimizer set to 0.8. Following the warm-up training, the learning rate is set to 0.01. The model is then trained for a total of 150 epochs. The result comprises the mean of 4 successive predictions, demonstrating the effectiveness and robustness of this methodology.

4.4 Comparisons

To evaluate the effectiveness and efficiency of our proposed defect detection model, we designed a series of comparative experiments.

Table 2 presents the evaluation results of our model on the NEU-DET dataset. the comparison results of class mAP, average mAP, and FLOPs(G) are shown. The optimal and second-best results in each line are highlighted in bold.

From Table 2, our model has achieved an overall performance increase of 4.4% in mAP compared to the basic model, YOLOv8, which can be speculated to be due to the efficiency of the FSF module.

For few-shot defect detection, this paper conducted few-shot training using 3/8 of the original training data as our training set. Furthermore, we compared the results of our model with YOLOv5 and YOLOv8. The comparative results of the few-shot experiment are presented in Table 3. The optimal results in each line are highlighted in bold.

The result revealed comprehensive improvements in our model's performance on the few-shot training set as well. In detail, the mean average precision (mAP) for the majority of categories is significantly superior to that of YOLOv8, thereby providing substantial evidence of the efficacy of our method in the context of few-shot defect detection. It is noteworthy that the category of Scratches, which originally exhibited a higher mAP in YOLOv8, experienced a

reduction in mAP when subjected to our model. This decrement may be attributed to the introduction of an excessive amount of irrelevant semantic information in the generated images for this particular category.

4.5 Ablation study

In order to verify the impact of each component in the controllable generation module and the effect of simulated feature fusion in the object detection module, this paper designed ablation experiments from the perspectives of image generation and detection networks, respectively, and demonstrated the effectiveness of this synthetic data augmentation approach.

4.5.1 Image generation

To demonstrate the effectiveness of the controllable generative network based on Stable Diffusion, this paper conducted ablation experiments on the LoRA and ControlNet.

For LoRA of the controllable generation module, in the comparative experiments based on the "generate2" generation scheme, LoRA weights are changed from 0 to 1 when generating images. Examples of the generated images are shown in Fig. 5. As LoRA weight decreases, the style becomes more random and unrealistic. Additionally, the detection network was trained using two different methods: directly augmenting the dataset with the generated and original images, and the improved method of synthesizing feature fusion. The results are presented in Table 4.

YOLOv8(1+1) represents the augmentation of one generated image set directly into the training set. Our(1+1) denotes the utilization of an enhanced object detection network that integrates features from both the original images and one generated images set during training. The best result in each column is highlighted in bold. From the results, it can be observed that as the LoRA weight decreases, the network learns more unrealistic information, leading to a decrease in mAP.

Table 4 The mAP comparison between mAP of different LoRA weight

LoRA weight	YOLOv8(1+1)	Our(1+1)
1.0	**78.2%**	**79.4%**
0.9	76.5%	78.8%
0.8	72.2%	77.0%
0.7	71.1%	75.2%
0.6	68.5%	73.4%
0.5	67.2%	72.8%
0.2	67.7%	73.5%
0.0	67.4%	72.4%

For the ControlNet, the following four models were selected, HED, Midas, Canny [43], and Lineart, to obtain outline maps of the images. ControlNet was employed to assist in generating images with different weights. (1) The Holistically-Nested Edge Detection (HED) model by Saining Xie et al. is designed to create edge maps using a nested approach. (2) Ranftl et al's Midas model estimates the depth map from a single image. (3) The Canny algorithm serves as a foundational edge detection method. (4) The lineart model excels at extracting line drawings from images. They are all models that generate contour maps

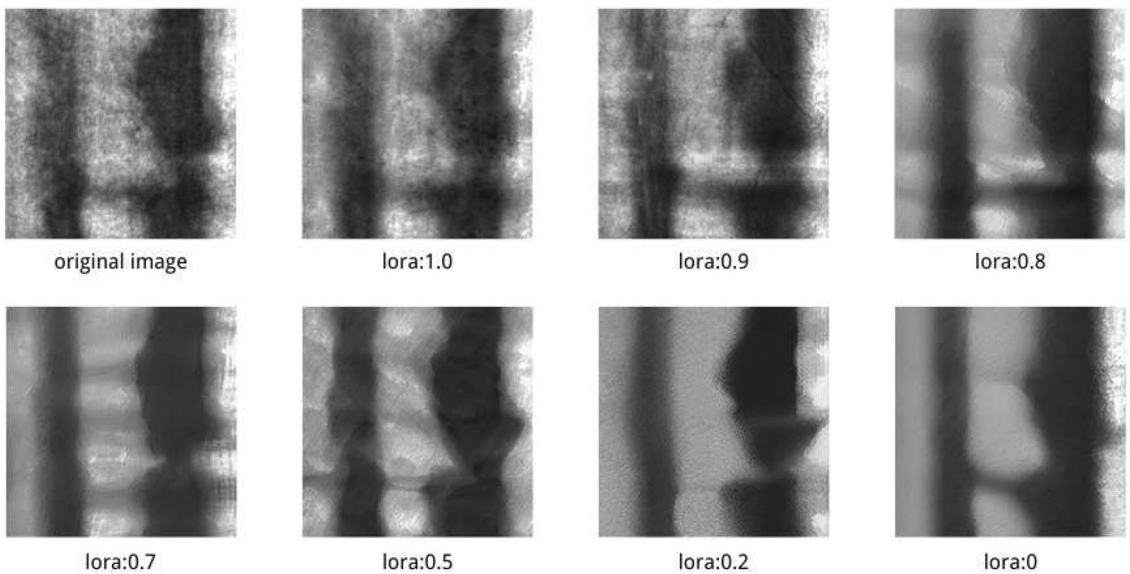

Fig. 5 The comparison results between the original image and the generated images at different LoRA weights

from the original image. The generated image samples are illustrated in Fig. 6.

It can be observed that under the control of different outline maps, the generated results exhibit a certain degree of control. The fixed texture lines in Canny lead to a lack of texture variation in the generated results. The excessive noise points in the outline map of Lineart result in both the generated images and the outline maps containing a large number of fixed noise points. On the other hand, the soft outlines of HED and Midas are well-suited for the NEU-DET defect dataset, enabling control over the defect positions while allowing for more variations.

4.5.2 Detection network

For the improved object detection network, experiments were conducted using the original images paired with a single generated image, as well as the original images paired with 4 different scales generated images. These experiments involved training the network using two methods: directly augmenting the training set and enhancing training through feature synthesis using our method. The experimental results are presented in Table 5.

From Table 5, YOLOv8(1+1) represents the augmentation of one generated image set directly into the training set. Our(1+1) denotes the utilization of an enhanced object detection network that integrates features from both the original images and one generated image during training. YOLOv8(1+4) represents the augmentation of 4 generated

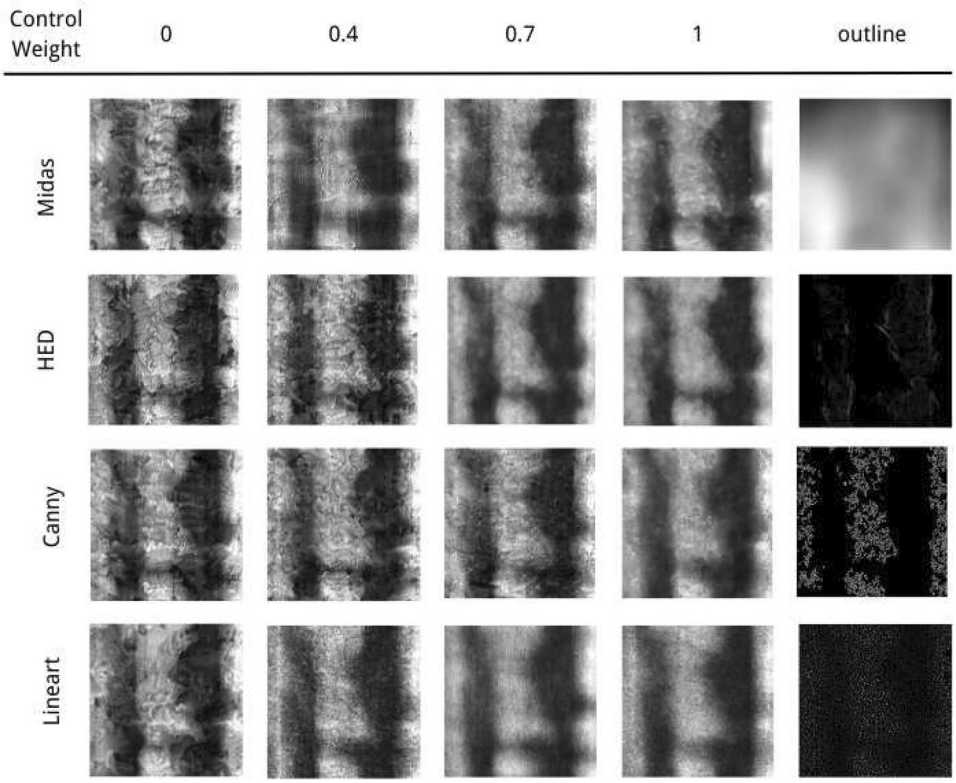

Fig. 6 Outlines and the images generated by HED, Midas, Canny, and Lineart models at different ControlNet weights

Table 5 The comparison between mAP of different training plans

	YOLOv8	YOLOv8 (1+1)	YOLOv8 (1+4)	Ours (1+1)	Ours (1+4)
Crazing	50.5%	53.3%	49.9%	**57.3%**	54.4%
Inclusion	79.9%	**81.8%**	76.9%	81.7%	**84.6%**
Patches	91.9%	91.2%	91.9%	**93.0%**	**93.6%**
Pitted surface	81%	79.2%	78.5%	**81.7%**	**86%**
Rolled-in scale	68.7%	68.4%	66.9%	**69.5%**	**77.5%**
Scratches	95.1%	**95.3%**	93.2%	93.4%	**95.6%**
mAP	77.9%	78.2%	76.2%	**79.4%**	**82.3%**

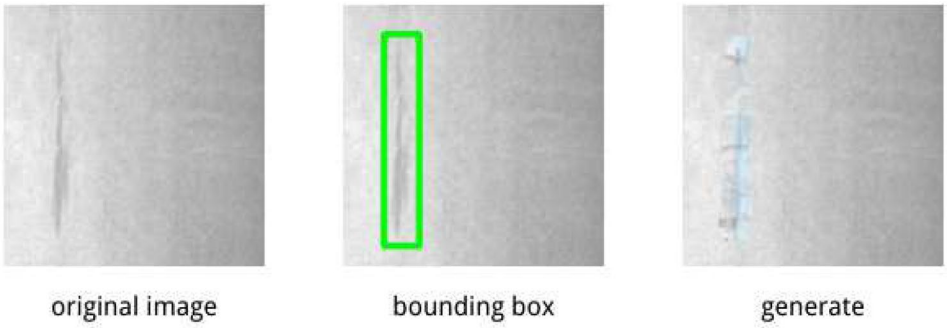

Fig. 7 The generated results of target-background separation

image sets directly into the training set. Our(1+4) denotes the utilization of an enhanced object detection network that integrates features from both the original images and 4 generated images during training. The best result in each line is highlighted in bold. Through ablation learning, it is evident that by utilizing a combination of the original dataset and a generated dataset for training, Favorable results were achieved with the baseline YOLOv8 network (YOLOv8(1+1)). Furthermore, further improvements were obtained by utilizing the enhanced YOLO-SD (YOLO-SD (1+1)). The best results were obtained when 4 generated datasets were fused using YOLO-SD (1+4). It is worth noting that when the original network was directly expanded with 4 generated datasets (YOLOv8 (1+4)), the accuracy decreased. Through analysis, this decline is attributed to the excessive reconstruction level of the generated results in the latter groups. Therefore, it is only through the improved YOLO-SD network that the impact of highly reconstructed images can be effectively harnessed during the training process.

4.6 Discussion

A detailed discussion of the experimental results is provided, analyzing the strengths and limitations of the approach. It was found that by introducing Stable Diffusion for generating simulated defect data and utilizing a multi-scale feature fusion network, the method effectively leverages a small amount of real data and a large amount of simulated data to improve object detection performance. However, it is also recognized that the quality of simulated data significantly impacts the model's performance. Therefore, careful control of the generation algorithms and parameters is necessary to ensure accurate simulation of real defect images.

A method was also explored in which the target region and background region of an image are separated to generate simulated images, and different reconstruction degrees are applied to each. As shown in Fig. 7, the background region is given a smaller degree of reconstruction, while the target region is given a larger degree of reconstruction. Finally, the generated results are stitched back into a single image. However, after multiple experiments, this method was deemed unsatisfactory. It introduces a noticeable boundary between the target and background images, which can lead to overfitting issues as the model tends to rely on this boundary as a decision criterion.

5 Conclusion

In this paper, a method based on YOLOV8 and Stable Diffusion is proposed to enhance the performance of steel surface defect detection. By utilizing Stable Diffusion to generate simulated defect data and an improved version of the YOLOv8 object detection framework, the challenges of data scarcity and time-consuming data annotation are effectively addressed, thereby improving the accuracy and generalization capability of object detection. The practical applicability of this approach in industrial scenarios is of significant value.

However, It is acknowledged that there is room for further improvement. Future research can explore ways to enhance the controllability of the generative network and investigate the integration of the generative network and the detection network into a single network. This integration would eliminate redundant data generated during synthesis, directly generate varying simulated features, and directly fuse them with the original features to achieve synthetic data augmentation. Additionally, conducting experiments on other defect datasets can further enhance detection performance.

Acknowledgements This work was supported by the Regional Innovation and Development Joint Fund of NSFC (No. U22A20167) and National key research and development program of China (No. 2021YFB3300503).

Data availability The datasets analyzed during the current study are available in the Baidu AIstudio repository, https://aistudio.baidu.com/datasetdetail/195425/1.

References

1. Liu Y, Gao X, Wen Z, Luo H (2023) Unsupervised image anomaly detection and localization in industry based on self-updated memory and center clustering. IEEE Transactions on Instrumentation and Measurement
2. Cao Y, Wan Q, Shen W, Gao L (2022) Informative knowledge distillation for image anomaly segmentation. Knowl-Based Syst 248:108846
3. Gao Y, Li X, Wang XV, Wang L, Gao L (2022) A review on recent advances in vision-based defect recognition towards industrial intelligence. J Manuf Syst 62:753–766
4. Gao Y, Gao L, Li X (2020) A generative adversarial network based deep learning method for low-quality defect image reconstruction and recognition. IEEE Trans Industr Inf 17(5):3231–3240
5. Jiao L, Dong S, Zhang S, Xie C, Wang H (2020) Af-rcnn: An anchor-free convolutional neural network for multi-categories agricultural pest detection. Comput Electron Agric 174:105522
6. Cha Y-J, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O (2018) Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Computer-Aided Civil and Infrastructure Engineering 33(9):731–747
7. Hou S, Dong B, Wang H, Wu G (2020) Inspection of surface defects on stay cables using a robot and transfer learning. Autom Constr 119:103382
8. Li Y, Xu, J (2020) Electronic product surface defect detection based on a mssd network. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 1, pp. 773–777. IEEE
9. Xing J, Jia M (2021) A convolutional neural network-based method for workpiece surface defect detection. Measurement 176:109185
10. Li M, Wang H, Wan Z (2022) Surface defect detection of steel strips based on improved yolov4. Comput Electr Eng 102:108208
11. Zhuxi M, Li Y, Huang M, Huang Q, Cheng J, Tang S (2022) A lightweight detector based on attention mechanism for aluminum strip surface defect detection. Comput Ind 136:103585
12. Ying Z, Lin Z, Wu Z, Liang K, Hu X (2022) A modified-yolov5s model for detection of wire braided hose defects. Measurement 190:110683
13. Li Y, Ni M, Lu Y (2022) Insulator defect detection for power grid based on light correction enhancement and yolov5 model. Energy Rep 8:807–814
14. Gao L, Zhang J, Yang C, Zhou Y (2022) Cas-vswin transformer: A variant swin transformer for surface-defect detection. Comput Ind 140:103689
15. Guo Z, Wang C, Yang G, Huang Z, Li G (2022) Msft-yolo: Improved yolov5 based on transformer for detecting defects of steel surface. Sensors 22(9):3467
16. Li L, Niu Z (2022) Few-shot tumor detection via feature reweighting and knowledge transferring. In: Proceedings of 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021), pp. 2606–2615 . Springer
17. Wu J, Liu S, Huang D, Wang Y (2020) Multi-scale positive sample refinement for few-shot object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16, pp. 456–472. Springer
18. Cheng J, Guo B, Liu J, Liu S, Wu G, Sun Y, Yu Z (2021) Tl-sdd: A transfer learning-based method for surface defect detection with few samples. In: 2021 7th International Conference on Big Data Computing and Communications (BigCom), pp. 136–143. IEEE
19. Wang H, Li Z, Wang H (2021) Few-shot steel surface defect detection. IEEE Trans Instrum Meas 71:1–12
20. Zoph B, Cubuk ED, Ghiasi G, Lin T-Y, Shlens J, Le QV (2020) Learning data augmentation strategies for object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16, pp. 566–583. Springer
21. Siu C, Wang M, Cheng JC (2022) A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection. Autom Constr 137:104213
22. Zhai G, Narazaki Y, Wang S, Shajihan SAV, Spencer BF Jr (2022) Synthetic data augmentation for pixel-wise steel fatigue crack identification using fully convolutional networks. Smart Struct Syst 29(1):237–250
23. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893. Ieee
24. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60:91–110
25. Bay H, Tuytelaars T, Van Gool L (2006) Surf: Speeded up robust features. Lect Notes Comput Sci 3951:404–417
26. Shumin D, Zhoufeng L, Chunlei L (2011) Adaboost learning for fabric defect detection based on hog and svm. In: 2011 International Conference on Multimedia Technology, pp. 2903–2906. IEEE
27. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: Relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1199–1208
28. Lai N, Kan M, Han C, Song X, Shan S (2020) Learning to learn adaptive classifier-predictor for few-shot learning. IEEE transactions on neural networks and learning systems 32(8):3458–3470
29. Wang R-Q, Zhang X-Y, Liu C-L (2021) Meta-prototypical learning for domain-agnostic few-shot recognition. IEEE Transactions on Neural Networks and Learning Systems 33(11):6990–6996
30. Ma Y, Bai S, Liu W, Wang S, Yu Y, Bai X, Liu X, Wang M (2021) Transductive relation-propagation with decoupling training for few-shot learning. IEEE transactions on neural networks and learning systems 33(11):6652–6664
31. Song K, Yan Y (2013) A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Appl Surf Sci 285:858–864
32. Guan Q, Chen Y, Wei Z, Heidari AA, Hu H, Yang X-H, Zheng J, Zhou Q, Chen H, Chen F (2022) Medical image augmentation for lesion detection using a texture-constrained multichannel progressive gan. Comput Biol Med 145:105444
33. Waheed A, Goyal M, Gupta D, Khanna A, Al-Turjman F, Pinheiro PR (2020) Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection. Ieee Access 8:91916–91923
34. Han C, Kitamura Y, Kudo A, Ichinose A, Rundo L, Furukawa Y, Umemoto K, Li Y, Nakayama H (2019) Synthesizing diverse lung nodules wherever massively: 3d multi-conditional gan-based ct image augmentation for object detection. In: 2019 International Conference on 3D Vision (3DV), pp. 729–737. IEEE
35. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695
36. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J et al (2021) Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR
37. Karimi Mahabadi R, Henderson J, Ruder S (2021) Compacter: Efficient low-rank hypercomplex adapter layers. Adv Neural Inf Process Syst 34:1022–1035
38. Xie S, Tu Z (2015) Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1395–1403

39. Ranftl R, Lasinger K, Hafner D, Schindler K, Koltun V (2020) Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Trans Pattern Anal Mach Intell 44(3):1623–1637
40. Zhang L, Agrawala M (2023) Adding conditional control to text-to-image diffusion models. arXiv preprint arXiv:2302.05543
41. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768
42. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: Convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19
43. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 6:679–698