mathematics

MDPI

Article

A Deep-Learning-Based Method for High-Precision Real-Time Detection of Steel Surface Defects

Guanying Song [1], Xiwen Wang [2], Gaoxia Fan [1], Hanquan Zhang [1], Guo Li [2], Zhenni Li [1], Jingyi Liu [3] and Dong Xiao [1,*]

[1] School of Information Science and Engineering, Northeastern University, Shenyang 110819, China; 20233361@stu.neu.edu.cn (G.S.); 2010292@stu.neu.edu.cn (G.F.); zhanghq12@mails.neu.edu.cn (H.Z.); lizhenni@ise.neu.edu.cn (Z.L.)
[2] School of Metallurgy, Northeastern University, Shenyang 110819, China; 20233273@stu.neu.edu.cn (X.W.); 20233462@stu.neu.edu.cn (G.L.)
[3] School of Science, Northeastern University, Shenyang 110819, China; liujingyi@mail.neu.edu.cn
* Correspondence: xiaodong@ise.neu.edu.cn

Abstract

Steel defects, stemming from issues like raw material imperfections and processing inconsistencies, present substantial challenges for the material's effective use and subsequent manufacturing. Consequently, the real-time, accurate, and rapid detection of these defects is paramount in production, playing a vital role in cost reduction, efficiency enhancement, and resource conservation. To address these needs, this paper proposes a deep deep-learning-based image recognition method for defect detection using YOLOv7 (You Only Look Once), designated YOLOv7-SGS. This approach introduces a novel architecture, the YOLOv7-SGS network, which builds upon the standard YOLOv7. The enhancements include integrating a Shape-IoU model into the core backbone, innovatively incorporating an SGE attention mechanism, and refining the convolution algorithm with GSConv to boost model performance. The resulting YOLOv7-SGS model achieves an absolute 6% improvement in mAP@0.5 compared to the baseline model. Moreover, it attains a detection speed of 32 FPS, showcasing significant advantages and offering valuable insights for future research and practical applications.

Keywords: steel defects; deep learning; image recognition; YOLOv7; attention mechanism

MSC: 68T07

Check for updates

Academic Editors: Wei Fang, Victor S. Sheng, Yanjie Fu, Jie Wen and Marjan Mernik

Received: 28 October 2025
Revised: 11 January 2026
Accepted: 3 February 2026
Published: 10 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Steel and its derivatives are integral to numerous facets of human life and have significantly propelled human civilization. In steel production, defect detection is indispensable. Identifying flawed materials early is crucial to prevent waste in subsequent manufacturing stages. Traditional inspection methods primarily involve manual visual checks and techniques employing infrared or eddy current technologies. However, manual inspection is often compromised by complex factory conditions, harsh environments, and operator visual fatigue after extended periods, all of which diminish the efficiency and accuracy of defect identification. Furthermore, the costs associated with initial training and labor impede overall productivity. While infrared and eddy current-based methods offer alternatives, their application is typically limited to smaller areas, rendering them unsuitable for inspecting large volumes of steel moving rapidly on production lines. Therefore, to

satisfy the requirements of large-scale, high-speed, and high-efficiency manufacturing, more effective and precise defect detection methodologies are imperative.

With their inherent strengths in automated learning and decision-making, deep-learning algorithms have emerged as a premier choice for defect detection. A well-trained recognition model can substantially enhance the speed of steel defect detection while maintaining superior accuracy over prolonged operational periods. Furthermore, training with extensive datasets enables these models to identify a broader range of defect types and adapt to diverse production environments. Critically, machine-based recognition involves minimal operational expenses, primarily limited to initial training costs.

The application of deep learning in steel defect detection heavily relies on advancements in Convolutional Neural Networks (CNNs). Li et al. [1] modified the YOLO (You Only Look Once) network into a fully convolutional architecture, enabling the trained model to predict both the location and dimensions of steel surface defects. In 2020, He et al. [2] introduced a classification-priority network combined with multi-group CNNs for detecting steel surface defects. The evolution of deep-learning frameworks has also been significant. For instance, in 2018, Chen et al. [3] developed a deep-learning framework employing a CNN and a Naive Bayes data fusion strategy to analyze cracks in single video frames. Their proposed NB-CNN demonstrated a superior hit rate and faster processing speeds compared to existing methods, even effectively identifying micro-cracks imperceptible to the human eye under conditions of low contrast and uneven illumination. In 2019, He et al. [4] presented CAE-SGAN, a semi-supervised learning approach for steel surface defects, to tackle the common problem of largely unlabeled sample images. CAE-SGAN begins by training a stacked Convolutional Autoencoder (CAE) model with extensive unlabeled data. To accommodate diverse defect morphologies, the algorithm incorporates a passthrough layer, enhancing the CAE's capacity for fine-grained feature extraction. Post-training, the CAE's encoder network functions as a feature extractor, and a softmax layer is integrated to form a novel classifier. To further enhance generalization, a Generative Adversarial Network (SGAN) is employed for semi-supervised learning, allowing the classifier to train on both real-world production line images and synthetically generated samples from SGAN. This semi-supervised method improved the detection accuracy for hot-rolled strip surface defects by approximately 16%. Also, in 2019, Fu et al. [5] introduced a compact and efficient CNN model based on a pre-trained SqueezeNet. This model emphasized the training of low-level features and integrated multiple receptive fields to achieve rapid and accurate classification of steel surface defects. The same year, He et al. [2] developed another novel defect detection method using a multi-layer feature fusion network. This network generates Regions of Interest (ROIs) through multi-level feature extraction, which are then processed by classifiers and bounding box regressors to predict defect types and locations. This approach exhibited higher accuracy and stability than traditional methods.

Addressing the persistent deficiencies and challenges within the steel industry, this study investigates a high-precision, real-time detection method for steel surface defects, leveraging deep learning and image recognition. The primary contributions of this research are as follows:

Methodological Innovation: To address the challenges posed by extensive background interference in images and significant variations in defect sizes, this paper introduces a novel architecture based on YOLOv7, namely the YOLOv7-SGS network. This enhancement integrates the Shape-IoU model into the core backbone and innovatively incorporates the SGE attention mechanism model. Additionally, the convolution algorithm is improved by utilizing GSConv to enhance the model's performance.

Experimental Validation: In comparison, the final YOLOv7-SGS model achieves a significant increase of 6% in mAP@0.5 over the baseline model. Furthermore, it attains a detection speed of 32 FPS, demonstrating notable advantages.

Application Value: For steel detection scenarios, YOLOv7-SGS is relatively lightweight and can be operated using simpler equipment. Its detection accuracy and detection rate are also capable of meeting the demands of actual production. This research also provides valuable insights for future studies.

2. Methodology

2.1. YOLOv7

The You Only Look Once (YOLO) network [6] is a single-stage object detection method renowned for its superior performance compared to other algorithms of comparable scale. In July 2022, Chien-Yao Wang et al. introduced YOLOv7, which has shown significant advantages over existing algorithms like YOLOX, YOLOv5 [7], DETR [8], SSD [9], and Faster R-CNN [10], and offers considerable potential for further enhancements.

The YOLOv7 network architecture [11] comprises three primary components: Input, Backbone, and Head. (The model's structure and components are illustrated in the accompanying figures.) YOLOv7 incorporates several upgrades over its predecessors, resulting in enhanced accuracy and speed. Its fundamental structure is depicted in Figures 1–3.

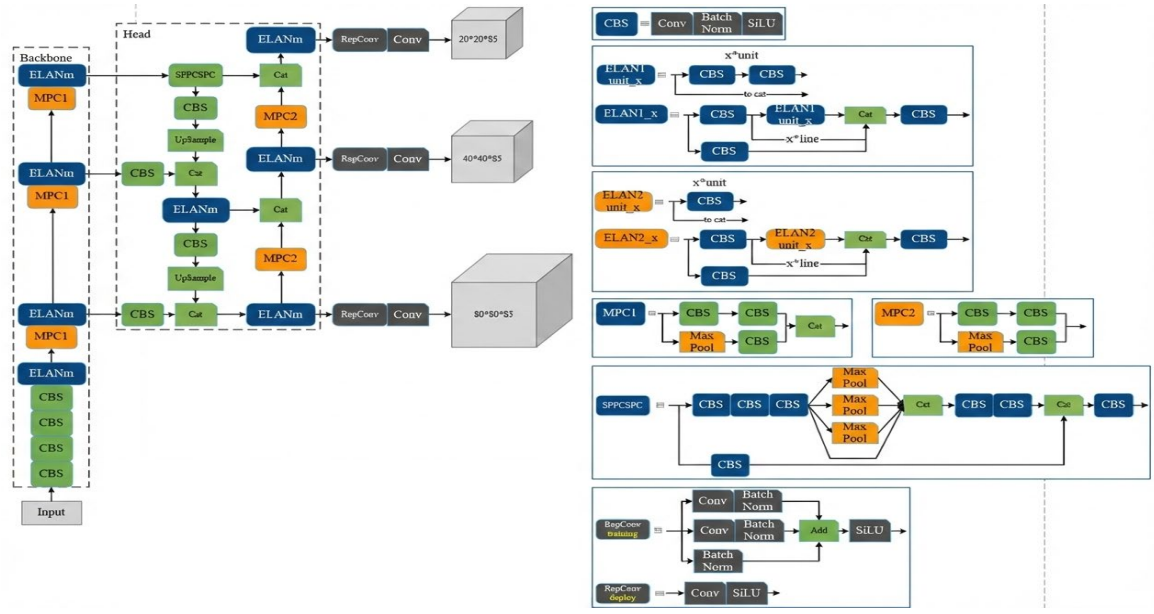


Figure 1. The Architecture of YOLOv7.

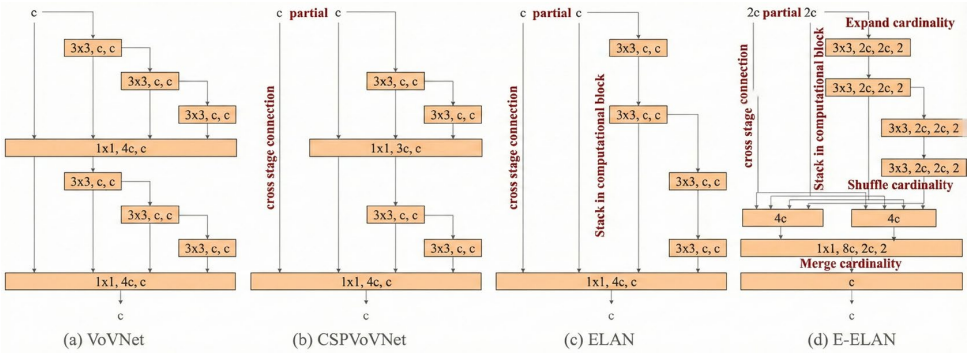


Figure 2. Special Modifications in YOLOv7.

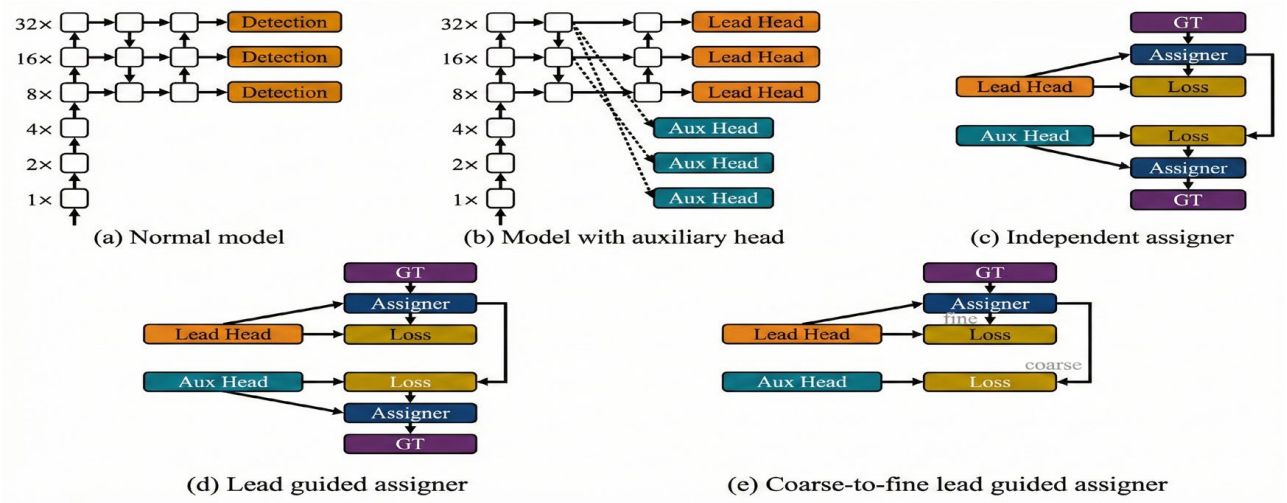


Figure 3. Implementation Principles of Each Module.

Input: The input stage encompasses two crucial components: data preprocessing and data augmentation. For preprocessing, YOLOv7 employs an adaptive image scaling technique [12]. This method intelligently resizes images to the target dimensions while preserving their original aspect ratio by dynamically calculating the optimal minimum padding. Compared to the traditional "letterboxing" approach, this significantly reduces ineffective, black-padded regions, thereby enhancing subsequent computational efficiency.

Regarding data augmentation, YOLOv7 incorporates an extensive suite of strategies. Beyond standard techniques like random horizontal flipping and color space adjustments (HSV), it innovatively introduces Mosaic augmentation, which combines contextual information from up to four training images. This method enhances data diversity while mitigating the creation of unnaturally distorted samples. These synergistic augmentation techniques effectively broaden the variety of training data and encourage the model to learn more robust feature representations.

Backbone: The YOLOv7 backbone module utilizes an enhanced architecture derived from CSPDarknet and incorporates the ELAN (Efficient Layer Aggregation Network) structure. This design improves feature extraction capabilities through more efficient gradient pathways, representing a significant optimization and innovation compared to the YOLOv5 [13] backbone.

Head: The Head component features an efficient multi-scale prediction network. Its primary function is to decode the features aggregated by the Neck module and generate the final object detection outputs. YOLOv7 employs an innovative extended ELAN (Efficient Layer Aggregation Network) architecture, specifically optimized to improve both detection accuracy and computational efficiency. The improved ELAN structure is illustrated as follows.

This architecture expands computational blocks using group convolution. While preserving the original ELAN transition layer structure, it innovatively incorporates operations such as channel expansion, shuffling, and cardinality merging. This design offers a dual benefit: it maintains network stability by ensuring smooth gradient flow and simultaneously enhances the network's feature learning capabilities without compromising the original optimization pathways.

In its feature processing pipeline, the enhanced ELAN architecture employs a hierarchical feature interaction mechanism. High-level features convey spatial detail via downsampling [14], while low-level features contribute semantic information through upsampling. This bidirectional feature fusion allows the network to acquire more comprehensive and discriminative representations across various scales, leading to a notable improvement in detecting objects of diverse sizes. This structure is particularly beneficial for small object detection, as it effectively aggregates multi-level detail features, thereby substantially increasing recognition accuracy. Figures 1–3 detail the architecture of the YOLOv7 network.

2.2. Proposed Method

Current improvements to the YOLO algorithm, such as Bi2F-YOLO, WG-YOLOv5, YOLOv7-Tiny, and Attention-YOLO, each have their own characteristics, as shown in Table 1.

Table 1. Brief Introduction of Each Model.

Model Name	Attention Mechanism Used	Lightweight Design	Target Domain	Core Innovation	Dataset
Bi2F-YOLO [15]	Dedicated attention mechanism not specifically mentioned	Base YOLOv7 architecture optimization	Underwater target detection	Proposed a dual-branch feature fusion framework to enhance feature extraction capability in underwater low-visibility scenarios	Underwater target-specific dataset
WG-YOLOv5 [16]	No dedicated attention mechanism	GSConv + Wavelet Transform	Wildfire real-time detection	Combined wavelet transform to enhance texture feature extraction, with GSConv reducing computational overhead	Wildfire scene-specific dataset
YOLOv7-Tiny [9]	No dedicated attention mechanism	Simplified ELAN structure + shallow feature fusion	Target detection for terminal devices (lightweight scenarios)	Streamlined network layers and optimized model parameter scale to adapt to edge computing	General object detection dataset (COCO subset)
Attention-YOLO [3]	CBAM (Channel + Spatial Attention)	Traditional convolution + attention lightweight adjustment	General defect detection	Introduced CBAM attention mechanism to enhance key feature weights, with no dedicated lightweight design	General industrial defect dataset

The primary enhancements proposed in this study are applied to the baseline YOLOv7 model. To improve the model's ability to identify defects that closely resemble the background (e.g., crazing) and larger defects (e.g., patches, scratches), an SGE (Spatial Group-wise Enhance) attention module is integrated only beneath the second MPConv layer within the Head. This integration strengthens feature extraction from input images while simultaneously making the model more lightweight and suitable for practical production environments [17]. Furthermore, all standard convolutions within the ELAN modules in the Head are replaced with GSConv. This modification reduces the computational load, contributing to a lighter model, and facilitates more comprehensive feature extraction. Additionally, the standard bbox_iou is replaced with Shape-IoU. This change enables the generated prediction boxes to align more accurately with the actual contours of the defects, thereby enhancing model precision and reducing errors such as false positives and incorrect merging of detections. The final refined architecture, designated YOLOv7-SGS, is illustrated in Figure 4, which shows the positions of the newly added SGE and GSConv modules within the YOLOv7-SGS framework.

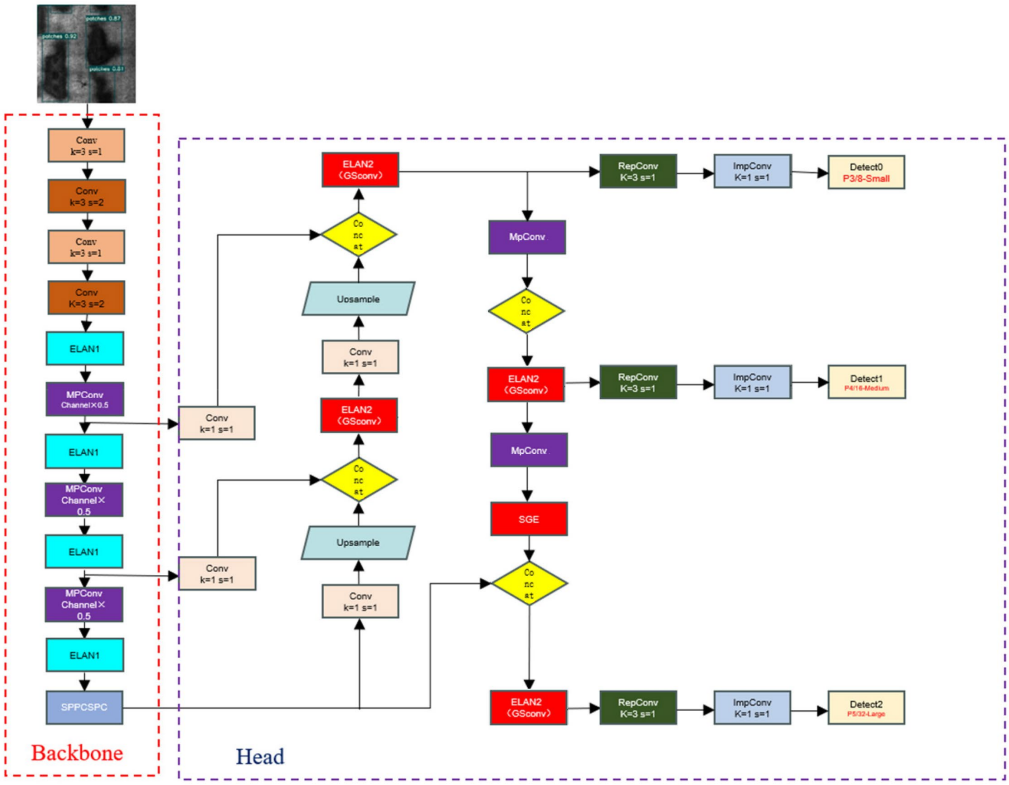


Figure 4. Enhanced Network Architecture.

2.2.1. Attention Model

Attention mechanisms, by simulating the human visual system, enable neural networks to focus on more critical information. In the field of computer vision, a series of efficient attention modules have been proposed, such as SENet, which focuses on channel relationships, and CBAM, which combines both channel and spatial attention. These mechanisms significantly enhance the representational capacity of convolutional neural networks by explicitly modeling feature dependencies. The SGE attention mechanism adopted in this paper is more suitable for steel detection tasks compared to traditional attention mechanisms.

Spatial Group-wise Enhance (SGE) is a lightweight attention mechanism introduced by Pan (2023) [18]. It aims to improve semantic feature learning in Convolutional Neural Networks (CNNs) via dynamic spatial grouping. The fundamental concept involves partitioning the channels of an input feature map into several groups. Within each group, a spatial attention mask is independently generated to amplify the semantic features of critical regions while suppressing noise. Specifically, an input feature map, X, is divided into G non-overlapping groups. For each group, global average pooling extracts a group-level global feature, g. This global feature is then used in a dot product with the local features at every spatial position within that group to compute similarity scores, thereby generating initial attention weights. These weights are subsequently standardized (by subtracting the mean and dividing by the standard deviation) and further refined using learnable scaling (γ) and offset (β) parameters. Finally, a Sigmoid function transforms these adjusted weights into attention masks, with values ranging from 0 to 1, which are then applied to scale the original feature values.

The SGE design substantially reduces computational complexity, with its parameter count being dependent only on the number of groups (G), typically 32 or 64, rendering it almost negligible. By leveraging the similarity between global and local features to guide attention, SGE dynamically enhances the feature responses of semantically relevant areas (e.g., defects on steel surfaces) while concurrently suppressing background noise. For instance, in shallower network layers, SGE tends to concentrate on local high-contrast regions. In contrast, in deeper layers, it integrates global contextual information across groups, thereby balancing multi-scale perceptual capabilities. This mechanism optimizes the spatial distribution of features without requiring intricate computations, making it highly suitable for the detection scenarios presented in this study. In the context of steel detection, compared to classical attention modules such as SENet and CBAM, this mechanism can achieve optimization of feature spatial distribution without requiring complex computations, demonstrating high feasibility in the detection scenario proposed in this paper. Its fundamental structure is depicted in Figure 5.

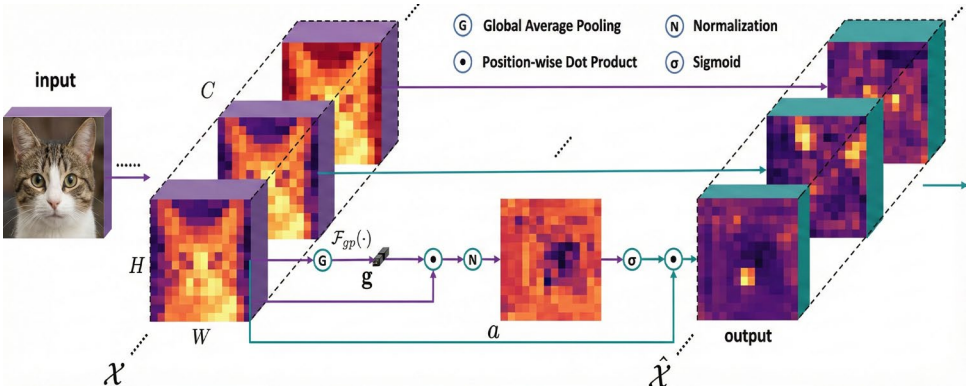


Figure 5. Brief Introduction Diagram of SGE Attention Mechanism.

2.2.2. GSConv

GSConv, introduced by Li et al. in 2024 [19], is an efficient convolutional module that significantly enhances the lightweight performance and defect detection capabilities of models like YOLOv7 by synergizing the strengths of group convolution and depthwise separable convolution. Its core design involves processing input feature maps in groups. Each group first undergoes an independent standard convolution to reduce parameters. This is followed by a depthwise separable convolution (which combines channel-wise spatial convolution with a 1×1 pointwise convolution) to decouple spatial feature extraction from channel information fusion. Concurrently, channel shuffling is introduced to improve inter-group communication. This approach reduces computational complexity while retaining the model's ability to represent multi-scale features. For steel surface defect detection, incorporating GSConv offers several advantages [20]. Firstly, replacing standard convolutional layers in YOLOv7 with GSConv can markedly decrease both parameter count and computational load, making the model better suited for real-time deployment on edge devices. Secondly, the lightweight design of GSConv mitigates the issue of semantic information loss often associated with traditional depthwise separable convolutions [21]. When combined with a feature shuffling mechanism, it can improve sensitivity to small defects, such as cracks and pits. These enhancements enable the detection model to achieve higher speeds without a substantial sacrifice in accuracy, aligning perfectly with this paper's objective of real-time steel defect detection on high-speed production lines. The operational principle of GSConv is illustrated in Figure 6.

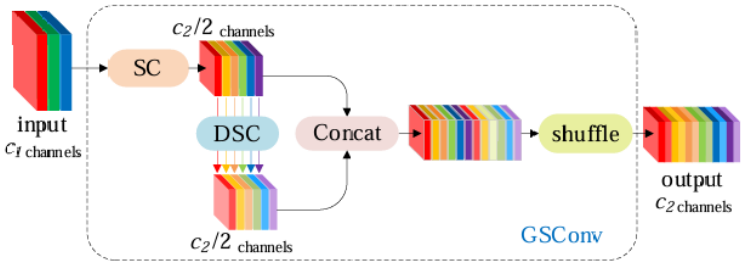


Figure 6. Brief Introduction Diagram of GSConv.

2.2.3. Shape-IoU Clustering

The precision of bounding box regression is crucial in both the training and prediction stages. This module is applied to training and inference tasks in this study. During training, accurate regression, guided by loss function optimization, positive sample matching, and effective feature extraction, enhances the model's ability to precisely localize defects, increases its sensitivity, and improves its discriminative power, leading to a better fit with ground truth. In the prediction phase, accurately regressed bounding boxes more closely conform to the actual contours of defects. This reduces instances of misclassifying defects in critical regions, incorrect merging of detections, erroneous suppression of valid boxes, and false retention of irrelevant boxes. Furthermore, it improves adaptability to varying input resolutions, thereby enhancing the localization accuracy and measurement reliability for defects.

Traditional Intersection over Union (IoU) is a key metric for assessing the overlap between predicted and ground truth bounding boxes. However, it solely considers the overlapping area of these rectangular boxes, neglecting the target's shape information. Since steel surface defects often exhibit irregular shapes, conventional IoU is insufficient for achieving the high-precision bounding box regression required. To address this limitation, this study employs Shape-IoU. Building upon the traditional IoU, Shape-IoU incorporates a shape similarity constraint. This allows for a tighter fit to irregularly shaped defects, reduces background interference, and consequently enhances the model's capability to detect shape-sensitive defects. From a mathematical perspective: Standard IoU, as a fundamental metric, relies solely on the intersection over union area ratio. It faces the gradient vanishing problem when there is no overlap and is insensitive to the shape alignment of bounding boxes. GIoU addresses the gradient issue in non-overlapping cases by introducing the minimum enclosing rectangle as a penalty term. However, its optimization remains relatively coarse; in containment scenarios, it can degenerate to IoU and lacks detailed consideration of geometric attributes.

Shape-IoU decouples the regression process geometrically, structurally integrating multiple dimensions such as angle alignment, center point distance, and aspect ratio similarity. This decomposed penalty mechanism endows the optimization process with directionality and order, significantly enhancing the perception and constraint capability for bounding box shapes, thereby achieving more precise and efficient localization convergence in complex scenes.

Mathematically, Shape-IoU is expressed as follows:

$$Shape - IoU = IoU \times \left(1 + \alpha \times S_f + \beta \times C_f\right) \quad (1)$$

where S_f is the shape factor, reflecting aspect ratio consistency, C_f is the scale factor, considering the influence of target size, α and β are balancing hyperparameters.

The shape factor is calculated as:

$$S_f = \exp(-\gamma \times |r_A - r_B|) \tag{2}$$

where r_A and r_B are the aspect ratios of the two bounding boxes, and γ is an adjustment coefficient.

The scale factor is calculated as:

$$C_f = \log\left(\frac{\min(w_A, w_B)}{\min(w_{img}, \mathrm{h}_{img})}\right) \times \log\left(\frac{\min(\mathrm{h}_A, \mathrm{h}_B)}{\min(w_{img}, \mathrm{h}_{img})}\right) \tag{3}$$

where w_A, h_A and w_B, h_B are the widths and heights of the bounding boxes, and w_{img}, h_{img} are the image dimensions.

The loss function is mathematically expressed as:

$$L_\{Shape - IoU\} \ = \ 1 - IoU + \alpha \times Distance_Loss + \beta \times Shape_Loss \tag{4}$$

Distance loss:

$$Distance\ Loss = \ \Sigma_\{i \in \{x, y\}\} \ (1 - e^{\{-\gamma \rho_i\}}) \tag{5}$$

where

$$\rho_x = (|c_x - c_x\hat{}*|) \ / \ (w^2 + (w\hat{}*)^2) \tag{6}$$

$$\rho_y = (|c_y - c_y\hat{}*|) \ / \ (h^2 + (h\hat{}*)^2) \tag{7}$$

γ is an adjustment parameter.

Shape loss:

$$Shape\ Loss = \ \Sigma_\{i \in \{w, \mathrm{h}\}\} \ \left(1 \ - \ e^{\{-\omega \theta_i\}}\right)^{\lambda} \tag{8}$$

where

$$\theta_w = (|w - w\hat{}|) / max(w, w\hat{}) \tag{9}$$

$$\theta_h = (|h - h\hat{}|) / max(h, h\hat{}) \tag{10}$$

ω and λ are shape adjustment parameters.

Specifically, the implementation of Shape-IoU within the YOLOv7 framework involves the following steps:

1. Derive shape features from rectangular dimension boxes: Calculate the aspect ratio features of each defect based on the dimensioned real- bounding box (rectangle). Although the NEU-DET dataset only provides rectangular labeling, the approximate shape tendency of defects can be captured through aspect ratio statistics. For each defect category, we calculate the aspect ratio distribution of its dimension frame as a prior knowledge of shape. The rationale for this approach is that similar defects tend to have similar morphological features (e.g., scratches are often elongated, patches are approximately square).

2. Modify the loss.py file in YOLOv7 by replacing the default BBoxIoU loss function used in the computeloss and computelossota functions. Embed Shape-IoU into the bounding box regression branch, enabling the training process to simultaneously optimize both rectangular box coordinates and shape similarity. The key parameters and their configurations are as follows in Table 2.

Table 2. Parameter Configuration and Rationale.

Parameter Value	Rationale	Reasons
α	0.5	Balances the penalty for shape discrepancy against that for center point distance.
eps	1×10^{-7}	Prevents division-by-zero errors.
β	0.5	Balance aspect ratio similarity loss
γ	2.0	Control the sensitivity of the center point distance loss
ω	2.0	Control the sensitivity of the difference in aspect ratio loss
λ	1.5	Nonlinear penalties for shape differences

3. When applying Non-Maximum Suppression (NMS) to the predicted boxes, incorporate the Shape-IoU score into the ranking process. This mechanism prioritizes the retention of candidate boxes that exhibit a high degree of shape correspondence with the ground truth. The pseudocode (Algorithm 1) is as follows:

Algorithm 1. Shape-IoU Enhanced Non-Maximum Suppression

Input: Model predictions P, confidence threshold θ_c, IoU threshold θ_i, scale parameter s
Output: Final detection results D
// Confidence Filtering
B ← {p ∈ P | p.conf > θ_c}
// Calculate Shape-Aware Score
for each b_i in B do
// Traditional score
S_trad ← b_i.obj_conf × b_i.cls_conf
// Shape consistency score
S_shape ← calculate_shape_consistency(b_i, B, s)
// Composite score
b_i.final_score ← 0.7 × S_trad + 0.3 × S_shape
end for
// Shape-Aware NMS
D ← shape_aware_nms(B, θ_i, s)
return D
There are the following definitions for functions that appear in pseudocode:
1. calculate_shape_consistency:
Parameter meaning:
b_i: The current bounding box, containing coordinates (x, y, w, h), confidence, etc.
B: A collection of all candidate boxes
s: Scale parameter, set to 0.2 in this study
Calculation process:
Calculate b_i Shape-IoU similarity to all other boxes in set B:

$$S_{shape}(b_i, b_j) = ShapeIoU(b_i, b_j)$$

The average of the highest similarity scores of the first K (K = 3) is taken:

$$shape_score = \frac{1}{K}\sum_{j=1}^{K} S_{shape}\left(b_i, b_j^{(top)}\right)$$

Normalization of scale parameters is applied

$$final_shape_score = exp(-s \times (1 - shape_score))$$

Algorithm 1. *Cont.*

Output:
Shape consistency score, range [0, 1]
2. shape_aware_nms:
Parameter meaning:
θ_i: The IoU threshold is set to 0.60 in this study
B: A collection of bounding boxes sorted by composite score
s: Scale parameter, set to 0.2 in this study
Calculation process:
Initialize the empty result list D
Loop when B is not empty:
Select the box b_max with the highest overall score in B and move it to D
Removed b_max from B
For each remaining box b_j in B:
Calculating the standard IoU: $iou = IoU(b_max, b_j)$
Calculating Shape-IoU: $shape_iou = ShapeIoU(b_max, b_j)$
Calculate the compound overlap measure: $overlap = 0.7 \times iou + 0.3 \times shape_iou$
If $overlap > \theta_i$, remove b_j from B
Back to D
Output:
Detection results after shape-sensing non-maximum suppression

The shape-aware intersection and union ratio proposed in this paper (Shape-IoU, implemented as shape_iou in code).

In this study, the input image size is 640 × 640, and the anchor strategy adopts: Anchor-based. Table 3 shows the sizes of the re-clustered anchor boxes.

Table 3. Sizes of Re-clustered Anchor Boxes.

Feature Map Size	Anchor Box Dimensions
80 × 80	[(12,16), (19,36), (40,28)]
40 × 40	[(36,75), (76,55), (72,146)]
20 × 20	[(142,110), (192,243), (459,401)]

3. Experiments and Discussion

3.1. Dataset Description and Preprocessing

This study is based on the publicly available NEU-DET steel surface defect dataset. The raw dataset contains 6 defect categories: Crazing, Inclusion, Patches, Pitted Surface, Rolled-in Scale, and Scratches. In order to focus on validating the effectiveness of the YOLOv7-SGS method in this paper, we selected the four most representative categories of defects with obvious differences in characteristics: Crazing, Inclusion, Patches, and Scratches for in-depth research, and at the same time, in real steel production quality inspection. These are also the four most common and most influential defects in material properties.

This study adopts a training–validation–test ternary split strategy. The specific division scheme is as follows: The training set comprises 960 images (accounting for 80%), with 240 images for each of the 4 defect categories, used for iterative model parameter training, hyperparameter tuning, and model selection. The validation set contains 120 images (accounting for 10%), with 30 images per defect category. The test set contains 120 images (accounting for 10%), with 30 images for each of the 4 defect categories. It is

kept entirely independent of the training process and is used solely for the final model performance evaluation. This approach allows for the allocation of relatively optimal data resources to the three critical stages of model development—"learning", "tuning", and "final evaluation"—under the constraint of limited data. It ensures both sufficient training and the rigor of the development process, as well as the credibility of the performance assessment. The splitting process strictly adheres to the principles of class balance and randomness: a fixed random seed is used to ensure the reproducibility of the results. There is no image overlap between the training and test sets, and the class distribution remains consistent with the original NEU-DET dataset (each defect category accounts for 25%). This avoids the impact of data distribution bias on the evaluation of the model's generalization ability. This study strictly guarantees the independence of the test data; the test results were not used for any hyperparameter tuning.

To achieve compatibility with YOLOv7, we standardized the dataset by accurately converting XML annotations into TXT files to ensure the effectiveness of model training.

The NEU-DET dataset provides annotations in XML format. However, YOLOv7 requires annotation information to be stored in text files (typically .txt format). Therefore, a conversion process was necessary to ensure compatibility. This conversion enables the effective utilization of the dataset with the YOLOv7 model. Maintaining the consistency and accuracy of annotation information throughout this transformation is critical. This ensures that the converted .txt annotation data is suitable for the YOLOv7 algorithm, thereby optimizing both training and inference phases.

YOLOv7 employs a comprehensive suite of data augmentation strategies to bolster model robustness [10] and generalization capabilities. These techniques include Mosaic augmentation (stitching four images), affine transformations, and color space perturbations. Mosaic augmentation, which intelligently combines four images and randomly resizes them, compels the model to learn multi-scale object detection concurrently, thereby significantly improving its adaptability to complex scenarios. Affine transformations simulate real-world variations in perspective through geometric distortions, supported by a dynamic bounding box calibration mechanism to maintain annotation accuracy. Moreover, YOLOv7 innovatively incorporates MixUp augmentation. This method linearly interpolates two images and their corresponding labels to create training samples with smoother decision boundaries, effectively mitigating overfitting. The model also leverages a Self-Adversarial Training (SAT) mechanism [10]. In its initial phase, SAT generates adversarially perturbed images; in the subsequent phase, this perturbed data is used to re-optimize the network parameters, enhancing the model's resilience to occlusions and noise. The synergistic application of these augmentation techniques generates highly diverse training samples, ultimately improving detection accuracy and adaptability. The parameters are shown in Table 4.

Table 4. Data Augmentation Specific Parameters.

Degrees	Translate	Scale	Shear	hsv_h	hsv_s	hsv_v	flipud	fliplr	mosaic	mixup
0	0.2	0.5	0	0.015	0.7	0.4	0	0.5	1	0

3.2. Experimental Setup

All algorithms in this study were implemented in Python, and experiments were conducted on an NVIDIA RTX 4060 GPU (NVIDIA, Santa Clara, CA, USA). To mitigate overfitting, data augmentation techniques, including image rotation and stitching, were applied to the input images. FPS measurements for all models follow a unified end-to-end inference protocol. The specific experimental environment and training parameters are detailed in Tables 5 and 6, respectively.

Table 5. Experimental Environment Configuration.

Component	Specification
CPU	Intel Core i9-13980HX @ 2.2 GHz
GPU	NVIDIA RTX 4060
Operating System	Windows 11
Deep-Learning Framework	pytorch 1.11.0
Programming Language	python 3.8
Memory (RAM)	32 GB RAM

Table 6. Training Hyperparameters.

Training Parameter	Value
learn rate	0.01
momentum	0.937
weight_decay	0.0005
image size	640 × 640
epochs	300
anchor strategy	anchor-based
batch size	32
optimizer type	SGD

3.3. Evaluation Metrics

This study primarily employs mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) and Frames Per Second (FPS) to evaluate detection accuracy and speed, respectively. The concepts of precision and recall are integral to understanding mAP. Precision quantifies the proportion of correctly identified positive instances (defects of a specific type) among all instances predicted as positive. Conversely, recall measures the proportion of actual positive instances that were correctly identified by the model. Precision and recall are defined by the following equations:

$$Precision = \frac{TP}{(TP + FP)} \tag{11}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{12}$$

In these categories, TP represents positively classified samples, FP refers to negatively classified samples mistakenly identified as positive, FN represents positively classified samples mistakenly classified as negative, and TN represents accurately classified negative samples.

3.4. Model Comparison Experiment

To establish a baseline, we selected three prominent object detection models for comparison: YOLOv5, YOLOX, and YOLOv7. All models were trained on our dataset using identical preprocessing steps and under the experimental conditions previously described. The trained models were then evaluated to assess their defect detection performance. The results are presented in Table 7.

Table 7. Impact of Different Models (Mean ± standard deviation over three runs. Due to architectural constraints and memory requirements, YOLOX could only operate at batch size 1 in this experimental environment, resulting in low FPS. The constant and all other models were measured with a batch size of 32. Therefore, the FPS values are not directly comparable).

	YOLOv5	YOLOX	YOLOv7
Crazing	0.311 ± 0.003	0.32 ± 0.004	0.326 ± 0.002
Inclusion	0.836 ± 0.005	0.732 ± 0.006	0.84 ± 0.003
Patches	0.895 ± 0.002	0.886 ± 0.003	0.895 ± 0.005
Scratches	0.874 ± 0.004	0.828 ± 0.005	0.875 ± 0.004
map@0.5	0.729 ± 0.006	0.692 ± 0.008	0.734 ± 0.005
FPS	35 ± 2	1 ± 0.1	27 ± 1

As evident from Table 8, YOLOv7 surpasses the other two baseline networks in terms of detection accuracy for steel surface defects, achieving an mAP@0.5 of 0.734. It consistently demonstrates higher Average Precision (AP) for each individual defect category compared to both YOLOv5 and YOLOX. Notably, for the "crazing" defect, YOLOv7 exhibits substantially superior detection accuracy.

Table 8. Ablation Study (Based on five independent runs, expressed as mean ± standard deviation).

	SGE	GSConv	Shape-IoU	mAP@0.5	FPS
YOLOv7	×	×	×	0.734 ± 0.005	26 ± 1
	√	×	×	0.76 ± 0.006	31 ± 2
	×	√	×	0.768 ± 0.007	29 ± 1
	×	×	√	0.746 ± 0.006	27 ± 1
	√	√	√	0.775 ± 0.005	32 ± 2

Regarding detection speed, while YOLOv7 is marginally slower than YOLOv5, its processing rate is still adequate for industrial production requirements. Furthermore, YOLOv7's significant accuracy advantage largely compensates for this slight speed difference [22]. On the contrary, due to its decoupled head body architecture, insufficient inference optimization (the default optimization of inference acceleration frameworks such as TensorRT is not supported, the implementation of activation functions and normalization layers is not efficient enough), memory limitations, and can only run at a batch size of 1, YOLOX's detection speed is greatly reduced, and the frame rate is much lower YOLOv5 and YOLOv7. To verify this result, we performed multiple experiments and the results stabilized between 0.9 and 1.1. It should be noted that the measured FPS of YOLOX in this experiment is only 1, which is based on a batch size of 1 and thus has limited reference value. A fair comparison of its FPS difference with the other two models would require testing under identical conditions on suitable hardware. However, since such an ideal environment for YOLOX may not always be available in practical applications—whereas the requirements for YOLOv7-SGS are easier to meet—this does not affect the conclusion that YOLOv7 remains a superior solution for real-time steel surface defect detection when both accuracy and speed are considered.

3.5. Experimental Evaluation of Model Improvements

To validate the effectiveness of our proposed method, YOLOv7-SGS, we conducted a series of comparative experiments under identical conditions. All experiments were based on five independent, complete training runs. Each run employed a pre-defined different random seed (e.g., [0, 42, 123, 2023, 2024]), which controlled the randomness in network weight initialization, data loading order, and all data augmentations. By recording the

mAP@0.5 value obtained from each run, we calculated the mean and standard deviation for this set of results. The final mAP@0.5 was reported with a standard deviation of ±0.006. The remaining comprehensive results are presented in Table 9.

Table 9. Experimental Parameter Settings.

Name	Parameter
Positive–Negative Sample Threshold	0.50
NMS Threshold	0.60
Confidence Threshold	0.25
Batch Size	32
Input Image	640 × 640
GPU	NVIDIA RTX 4060
CPU	Intel Core i9-13980HX @ 2.2 GHz

As shown in Table 10, anchor boxes generated through Shape-IoU clustering [23] exhibit a distinct advantage in steel defect detection. Compared to methods lacking this module, both mAP@0.5 and detection frame rate (FPS) showed improvement. This is primarily because Shape-IoU can better adapt to the geometric characteristics of defects on steel surfaces, thereby providing more task-specific prior box initialization. Furthermore, the integration of the SGE attention mechanism and the application of the GSConv convolution method [24] resulted in a noticeable enhancement in model performance. GSConv, as a lightweight convolutional structure, reduces parameter and computational redundancy while maintaining feature representation capability, thus accelerating the training and inference processes. Meanwhile, the SGE module effectively focuses on salient features of defect regions through its spatial group enhancement mechanism, compensating for potential subtle semantic losses that GSConv might introduce in deeper layers. The synergy between these two components achieves a balance between accuracy and efficiency. The synergistic fusion of these modules yielded the best detection results among the tested configurations, thereby validating the model's effectiveness. This confirms that our proposed modifications substantially refine the detection solution and enhance the practical application of image recognition for steel defect detection.

Table 10. Detection Accuracy Comparison at Different Positions (Single run, random seed = 42).

	After the First Mpconv	After the Second Mpconv
Crazing	0.320	0.326
Inclusion	0.817	0.840
Patches	0.770	0.895
Scratches	0.844	0.875
map@0.5	0.729	0.734
FPS	26	27

Regarding the placement of the SGE module, it is commonly positioned in the Head section. We tested detection accuracy and FPS by placing it after different Mpconv layers and found that placing it after the second Mpconv module resulted in better detection accuracy and higher FPS. This is because positioning the SGE attention mechanism at this stage effectively alleviates the issue of missing small targets. Embedding it into the second layer of MPConv (rather than deeper layers) optimizes information flow during the early feature extraction stage, avoiding efficiency degradation caused by complex computations in deeper networks.

To delve deeper into the impact of the SGE attention mechanism, we generated Grad-CAM (Gradient-weighted Class Activation Mapping) [25] visualizations for the baseline YOLOv7 model and for a YOLOv7-S (YOLOv7-SGE) model enhanced with the SGE module. This allowed us to observe the specific regions of focus for each network and evaluate the SGE attention mechanism's influence. The GradCAM visualizations are presented in Figure 7.

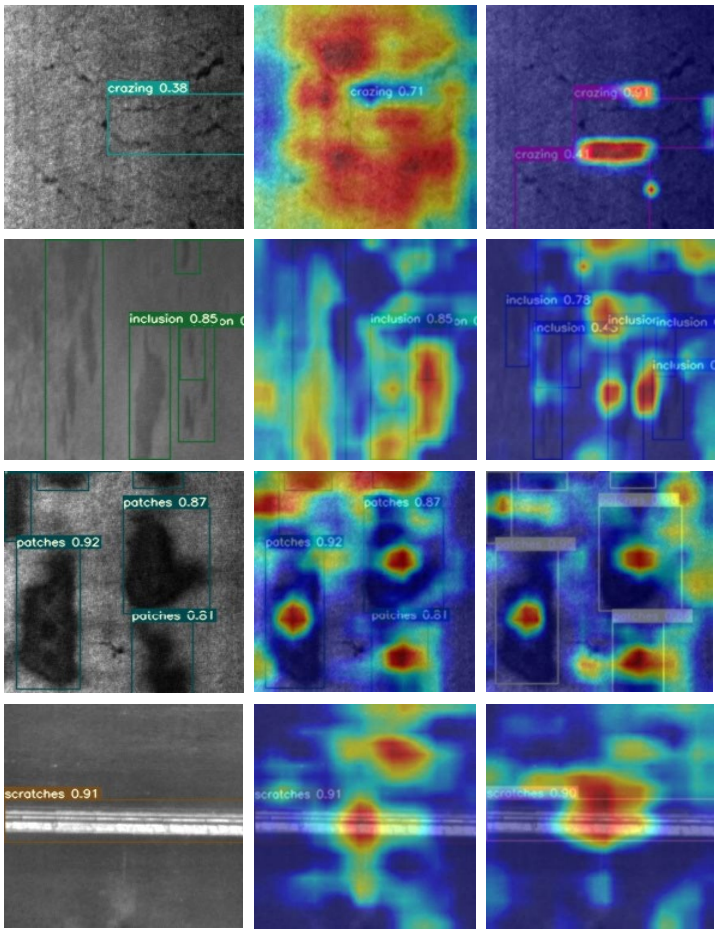


Figure 7. GradCAM visualizations for different defect models. For each defect type, the left image corresponds to the original YOLOv7, and the right to YOLOv7-S.

As illustrated in Figure 7, the SGE mechanism enhances the feature representation of specific semantic regions. Consequently, its integration strengthens the baseline network's ability to focus on defect features. For subtle defects that closely resemble the background, such as crazing, the SGE module effectively mitigates background interference, preventing the model from erroneously focusing on large, irrelevant background areas. This improvement significantly enhances the identification of targets that are visually similar to the background. In the case of larger defects, like scratches, the baseline network (without SGE), while detecting parts of the defect, often incorrectly focused on background regions where no defects were present. After incorporating the SGE module, the model demonstrably shifts its recognition focus to the actual defective locations. This not only ensures more complete coverage of the defect but also avoids misclassifying the background as a defect, thereby improving the recognition capability for larger defects. In summary, the integration of the SGE attention mechanism module represents a successful optimization.

To provide a more detailed and intuitive evaluation of the model's detection capabilities, we compared the detection accuracy of the baseline and enhanced models for different defect types. The comparison was conducted under the same parameter settings and experimental setup as used in the ablation studies. All experiments were based on five independent, complete training runs. Each run employed a pre-defined different random seed (e.g., [0, 42, 123, 2023, 2024]), which controlled the randomness in network weight initialization, data loading order, and all data augmentations. By recording the mAP@0.5 value obtained from each run, we calculated the mean and standard deviation for this set of results, yielding a final standard deviation of ±0.006 for mAP@0.5. The detailed comparative data are presented in Table 11.

Table 11. Model Detection Accuracy Comparison for Different Defect Types (Based on five independent runs, expressed as mean ± standard deviation).

Model	Crazing	Inclusion	Patches	Scratches	mAP@0.5	FPS	MParams	GFLOPs
YOLOv7	0.326 ± 0.005	0.84 ± 0.003	0.895 ± 0.001	0.875 ± 0.006	0.734 ± 0.005	26 ± 1	36.8	103.2
YOLOv7-SGS	0.429 ± 0.002	0.841 ± 0.004	0.922 ± 0.003	0.909 ± 0.003	0.775 ± 0.006	32 ± 2	34.8	97.6

As shown in Table 11, the enhanced model (YOLOv7-SGS) exhibits substantial improvements in detecting all defect types. Notably, detection accuracy for crazing, patches, and scratches saw particularly significant increases of 31.6%, 3.1%, and 3.9%, respectively. Furthermore, while the detection accuracy for 'inclusion' did not show a marked increase, it maintained its original high level. This indicates that the addition of modules did not compromise performance for this specific defect type but rather contributed to the overall anticipated enhancements. It is worth noting that the performance improvement for the crazing category is particularly significant (increasing from 0.326 to 0.429), while the detection results for the inclusion category remain largely unchanged. This discrepancy may stem from the following factors: (1) Crazing typically manifests as dense textures with low contrast against the background, and the added modules may have enhanced the model's perception of low-frequency textures and subtle differences; (2) The imbalance in sample distribution across different categories in the dataset may affect the model's feature learning for minority classes; (3) Inclusion defects vary widely in morphology and occur randomly in location, making feature localization more challenging, and the optimization effect of the existing modules on their feature extraction and localization is relatively limited. Figure further provides a detailed comparison of detection results for different defects on the test set through category-specific precision–recall curves, offering a more comprehensive view of the performance trends for each category from the evaluations of the baseline and improved models. Most encouragingly, the model's overall mAP@0.5 increased by 6%, which further confirms that the proposed modules possess unique advantages in the task of steel surface defect detection, demonstrating good application value. Figure 8 presents a detailed comparison of detection results for different defects from the test set, as evaluated by both the baseline and the improved models.

As shown in Figure 8, the original YOLOv7 model is prone to missed detections (false negatives) and false alarms (false positives) across various defect types, performing particularly poorly on crazing defects. In contrast, YOLOv7-SGS demonstrates superior performance in accurately identifying various defects. Notably, for crazing defects, the improved model achieves a significant accuracy boost of over 30%. For scratches, the original YOLOv7 often misidentified background regions as defects, a problem the improved model effectively mitigates. Detection speed also shows commendable improvement. For small defects that resemble the background, SGE provides considerable assistance, while for irregular boundaries, Shape-IoU plays a more significant role. In conclusion,

the enhanced YOLOv7-SGS model exhibits superior speed and accuracy for both small, background-indistinct defects and large, prominent defects.

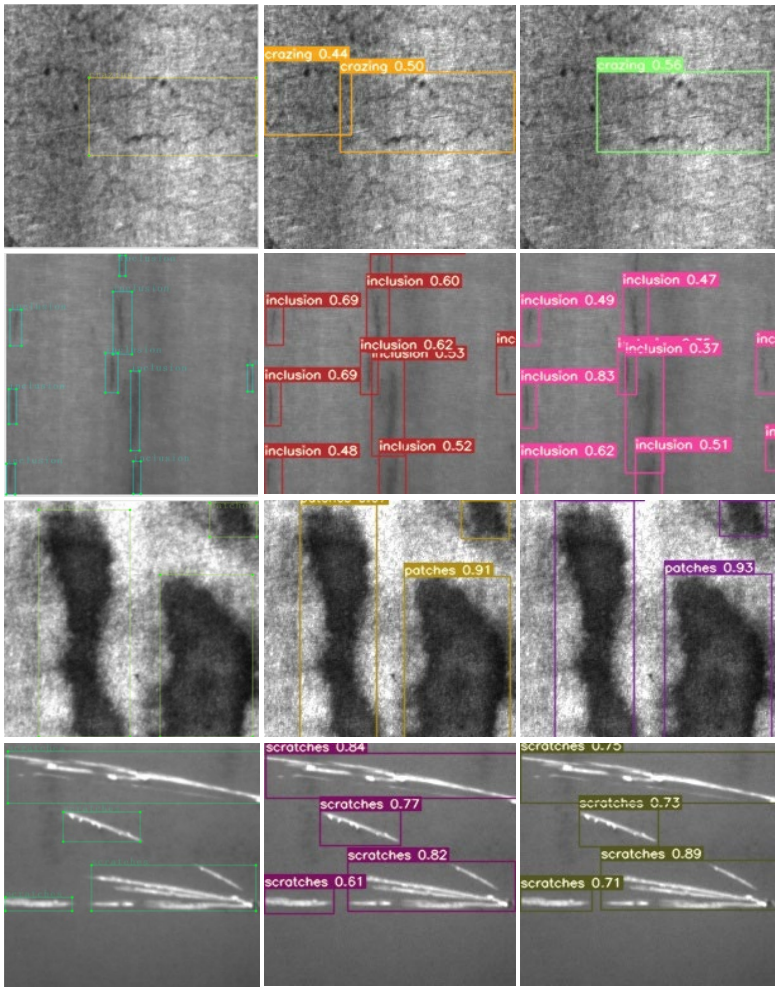


Figure 8. Detection results for four defect types by different models. Columns from left to right: Ground Truth, YOLOv7, and YOLOv7-SGS. Detected bounding boxes, class labels, and confidence scores are displayed on the images.

3.6. Further Comparison Experiment

To investigate whether the YOLOv7-SGS model offers more distinct advantages for steel detection and to provide a more comprehensive performance comparison, we introduced the DETR model based on the Transformer architecture, the latest lightweight version of the EfficientDet series renowned for its efficiency, and Faster R-CNN as a classic two-stage benchmark, in addition to the existing YOLO series. Under the same experimental conditions as described in Section 3.2, comparative experiments were conducted, yielding the results shown in Table 12.

Table 12. Comparison of detection accuracy of models on different defects (Based on three runs, expressed as mean ± standard deviation).

Model	mAP@0.5	FPS	Parameters	GFLOPs
YOLOv8n	0.741 ± 0.007	42 ± 3	3.1	8.2
DETR-ResNet50	0.698 ± 0.012	18 ± 2	41.2	86.5
EfficientDet-D0	0.712 ± 0.008	35 ± 2	3.9	2.5
Faster R-CNN	0.723 ± 0.009	15 ± 1	41.3	180.5
YOLOv7-SGS	0.775 ± 0.006	32 ± 2	34.8	97.6

The results show that our proposed YOLOv7-SGS model surpasses all compared models in mAP@0.5 while maintaining excellent performance in FPS, outperforming non-real-time architectures such as DETR and Faster R-CNN. Compared to the latest real-time detectors like YOLOv8, it achieves the best balance between speed and accuracy. Notably, although YOLOv8n has an extremely small parameter count and the fastest speed, its accuracy is slightly lower than our model, and it performs poorly on more difficult-to-recognize defects such as crazing. This demonstrates the comprehensive superiority of YOLOv7-SGS in the task of steel surface defect detection.

3.7. Data Augmentation Specialized Experiment

To investigate whether the data augmentation processing in YOLOv7 has substantive significance, experiments were set up starting with no augmentation, then progressively adding basic augmentation and various specialized augmentation schemes. We measured the throughput during training (training FPS), which is the number of training images that can be processed per unit time. This differs from inference FPS, which reflects the impact of additional computational overhead from data enhancement on training speed. These were used to compare the effectiveness of different data augmentation methods and measure their contributions. In addition, data enhancement is only applied during the training phase and does not affect the inference speed of the final deployed model. Model inference FPS (reported in Tables 7, 11 and 12) was measured with all data enhancements disabled. The regularization details are shown in Table 13.

Table 13. Regularization Details.

Dropout	Not Used
Batch Normalization	momentum: 0.937 weight_decay: 0.0005 warmup_epochs: 3.0 warmup_momentum: 0.8 warmup_bias_lr: 0.1
Early Stopping Strategy	Not Used

To quantify the degree of model overfitting, we define the Overfitting Index (OI) as follows:

$$OI = \left(\frac{L_{train} - L_{val}}{L_{train} + \epsilon}\right) \times \left(\frac{A_{train} - A_{val}}{A_{val} + \epsilon}\right)$$

Among them:

L_{train}: The final loss value of the training set
L_{val}: validates the final loss value of the set
A_{train} : The final mAP@0.5 of the training set
A_{val} : validation set final mAP@0.5
$\epsilon = 10^{-7}$: a numerical stability constant

If:

OI $\approx$ 0: The model is not overfitted, and the training and validation performance are consistent
OI > 0: There is an overfit, with higher values indicating more severe overfitting
OI < 0: There may be underfitting or measurement errors

The final experimental results are shown in Table 14 below:

Table 14. Experimental Data After Successively Adding Modules (Single run, random seed = 42).

Augmentation Combination	mAP@0.5	Recall	Precision	FPS	Overfitting Index
No Augmentation	0.701	0.682	0.745	34	1.28
Basic Augmentation Only (Flip + Color Disturbance)	0.738	0.721	0.783	33	0.95
+Mosaic Splicing	0.756	0.745	0.801	32	0.72
+Affine Transformation	0.772	0.763	0.818	30	0.55
+Self-Adversarial Training (SAT)	0.775	0.769	0.823	32	0.48

As can be seen, as the augmentation schemes increase, mAP@0.5, Recall, and Precision all increase to varying degrees and are positively correlated with the number of additions. FPS slightly decreases due to the gradually increasing complexity of the structure. This is because Mosaic stitching requires simultaneous processing of 4 images, Affine transform involves geometric calculations, and SAT requires additional forward–backward propagation; Complex data augmentation may make data preprocessing a bottleneck in the training pipeline. The enhanced operation of randomization reduces the locality of data access and affects caching efficiency. Despite this, the drop in training FPS is less than 5%, which is within an acceptable range, while the performance gain is significant. The overfitting index shows a more significant decrease, reflecting the effectiveness of the data augmentation schemes. Notably: Mosaic augmentation makes the greatest contribution: increasing mAP@0.5 by approximately 2.4% and significantly improving the model's multi-scale detection capability, making it an excellent data processing method.

3.8. Code Availability

The source code for implementing YOLOv7-SGS and reproducing all experiments is publicly available under the MIT License. The repository is hosted at: https://github.com/12344-netizen/yolov7-SGS (accessed on 6 January 2026).

To ensure exact reproducibility of the results reported in this paper, we provide a versioned release tagged v1.0.0-paper. The release tag link is: https://github.com/12344-netizen/yolov7-SGS/releases/tag/v1.0-paper (accessed on 6 January 2026).

This frozen version includes the complete training and evaluation scripts, configuration files, and pretrained model weights.

The dataset used in this study, NEU-DET, is publicly available at: http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/ (accessed on 6 January 2026).

4. Conclusions

This study focuses on the real-time detection of steel surface defects during steel production [25]. We began by comparing traditional detection methods and identified their significant limitations, prompting our exploration of deep-learning approaches. Using the NEU-DET dataset, we evaluated the accuracy and speed of baseline YOLO architectures (YOLOv5, YOLOX, and YOLOv7). Subsequently, we incrementally integrated SGE, Shape-IoU, and GSConv modules, assessing the impact of each optimization.

The YOLOv7-SGS model proposed in this study achieves significant performance improvements in the task of steel plate surface defect detection, with mAP@0.5 increasing by about 6%, detection speed reaching 32 FPS, and an about 23% improvement over the baseline. This performance enhancement primarily stems from the synergistic optimization of three modules: GSConv, the SGE attention mechanism, and Shape-IoU. The underlying mechanisms can be clarified through comparative analysis with relevant literature.

The GSConv module maintains sufficient semantic representation capability within a lightweight design. Compared to the widely adopted depthwise separable convolution, GSConv introduces grouped convolution and channel shuffling, reducing computational costs while preserving cross-channel feature interactions, thereby mitigating the common issue of fine defect feature loss in lightweight networks. This design aligns with the findings of Li et al. [19], who reported that using GSConv in general object detection led to reduced parameter counts while maintaining stable accuracy. In this study, the model's parameter count decreased by 20.2%, and detection accuracy for large-area patch defects (patches) improved by 3.1%. This can be attributed to the enhanced inter-group information flow enabled by channel shuffling, consistent with the observations of Luo et al. [24] in wildfire detection tasks.

The SGE attention mechanism provides more precise feature focusing on addressing the challenge of low contrast between steel defects and the background, making them difficult to distinguish. Compared to SENet [18], which relies solely on channel attention, or CBAM [17], which has higher computational complexity, SGE dynamically generates spatial attention masks within groups through dynamic spatial grouping, enhancing local defect feature responses at a lower computational cost. In this study, the AP value for crazing-type defects increased by 31.6%, corroborating the conclusion of Huilin et al. [18] that SGE exhibits stronger focusing capability for low signal-to-noise ratio features in bearing fault detection. Feature visualization results also show significantly enhanced saliency in defect regions, indicating that SGE effectively suppresses background interference.

The Shape-IoU loss function addresses the irregular morphology of steel defects, improving bounding box regression accuracy. Traditional IoU and its improved forms, such as DIoU, do not fully account for shape similarity, leading to inaccurate localization of irregular defects. Shape-IoU explicitly constrains the aspect ratio and size matching of bounding boxes by introducing shape and scale factors. In this study, the IoU values for all defect categories exceed 0.67, consistent with the trend reported by Zhao et al. [23] in similar industrial detection tasks, where shape-aware loss led to improved localization accuracy. Its characteristic of not relying on anchor box priors also enhances the model's adaptability to defects of varying morphologies.

This study still has several limitations. The recognition accuracy for inclusion-type defects, which are highly similar in morphology and background texture, did not significantly improve, reflecting the current attention mechanism's inadequacy in decoupling subtle features. Additionally, the evaluation framework could be further refined—for instance, by supplementing metrics such as mAP@[0.5:0.95] and F1-score to provide a more comprehensive performance assessment. Efficiency gains also need to be fully quantified with specific data, such as parameter counts and single-image inference times. While existing visualization results offer reference value, their conclusions still require more substantial quantitative data support. Finally, the model's performance currently relies heavily on sufficient annotated data. Under real-world industrial conditions where sample acquisition is challenging, its data efficiency and generalization capability need further enhancement.

Author Contributions: Conceptualization, G.S., J.L., and D.X.; methodology, X.W. and G.L.; software, G.S. and G.L.; validation, H.Z. and Z.L.; formal analysis, G.F. and Z.L.; investigation, G.S. and X.W.; writing—original draft preparation, G.S., X.W., and J.L.; writing—review and editing, G.F., H.Z., and D.X.; visualization, G.S., X.W., and G.L.; funding acquisition, D.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key R&D Program of China under Grant 2022YFB2703304; in part by the National Natural Science Foundation of China under Grant 52074064; in part by the Natural Science Foundation of Science and Technology Department of

Liaoning Province under Grant 2024-MSLH-524; in part by the Fundamental Research Funds for the Central Universities under Grant N25GFZ010.

Data Availability Statement: The data presented in this study are openly available in http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/ (accessed on 6 January 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dong, H.; Song, K.; He, Y.; Xu, J.; Yan, Y.; Meng, Q. PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7448–7458. [CrossRef]
2. He, Y.; Song, K.C.; Meng, Q.G.; Yan, Y.H. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [CrossRef]
3. Chen, F.C.; Jahanshahi, M.R. NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4392–4400. [CrossRef]
4. He, D.; Xu, K.; Zhou, P.; Zhou, D.D. Surface defect classification of steels with a new semi-supervised learning method. *Opt. Lasers Eng.* **2019**, *117*, 40–48. [CrossRef]
5. Fu, G.Z.; Sun, P.Z.; Zhu, W.B.; Yang, J.X.; Cao, Y.L.; Michael, Y.Y.; Cao, Y.P. A deep-learning-based approach for fast and robust steel surface defects classification. *Opt. Lasers Eng.* **2019**, *121*, 397–405. [CrossRef]
6. Pasupuleti, S.; Ramalakshmi, K.; Gunasekaran, H.; Arokiaraj, R.M.; Debnath, S.; Jebaseeli, T.J. An Enhancement of Object Detection Using YOLO V8 and Mobile Net in Challenging Conditions. *SN Comput. Sci.* **2025**, *6*, 321. [CrossRef]
7. Xu, L.; Dong, S.; Wei, H.; Ren, Q.; Huang, J.; Liu, J. Defect signal intelligent recognition of weld radiographs based on YOLO V5-IMPROVEMENT. *J. Manuf. Process.* **2023**, *99*, 373–381. [CrossRef]
8. Wang, H.; Zhang, C.; Wang, Y.; Ni, P.; Wang, Y. Segmentation of Inner Surface Defects of Stainless Steel Pipes Based on Semi-bilateral Efficient Self-Attention Network. *J. Nondestruct. Eval.* **2025**, *44*, 39. [CrossRef]
9. Zhijun, G.; Weiming, L.; Qiujie, C.; Hongbo, Z. Terminal strip detection and recognition based on improved YOLOv7-tiny and MAH-CRNN+CTC models. *Front. Energy Res.* **2024**, *12*, 1345574. [CrossRef]
10. Wang, B.; Liang, P.; Zhang, L.; Wang, X.; Yuan, X.; Zhou, Z. Enhancing robustness of cross-machine fault diagnosis via an improved domain adversarial neural network and self-adversarial training. *Measurement* **2025**, *250*, 117113. [CrossRef]
11. Zhu, J.; Qin, C.; Choi, D. YOLO-SDLUWD: YOLOv7-based small target detection network for infrared images in complex backgrounds. *Digit. Commun. Netw.* **2025**, *11*, 269–279. [CrossRef]
12. Ciulla, C.; Shabani, B.; Doda, G.; Hoxha, R.; Zaimaj, S. Two-dimensional image zooming by adaptive Sinc-based technique. *Innov. Emerg. Technol.* **2023**, *10*, 1–10. [CrossRef]
13. Xue, Y.; Wang, J.; Hu, S.; Zhang, Z.; Zhou, C.; Wang, S.; Xie, Z.; Wang, S. Persistent calyx of Rosa roxburghii recognition based on SimAM-YOLO-v5s. *J. Food Meas. Charact.* **2025**, *19*, 4715–4726. [CrossRef]
14. Wei, C.; Bao, Y.; Zheng, C.; Ji, Z. AMFNet: Aggregated multi-level feature interaction fusion network for defect detection on steel surfaces. *J. Intell. Manuf.* **2025**. [CrossRef]
15. Liu, X.; Zhao, K.; Liu, C.; Chen, L. Bi2F-YOLO: A novel framework for underwater object detection based on YOLOv7. *Intell. Mar. Technol. Syst.* **2025**, *3*, 9. [CrossRef]
16. Luo, L.; Wang, H. *WG-YOLOv5: Improved YOLOv5 Based on Wavelet Transform and GSConv for Real-Time Wildfire Detection*; Chongqing University of Posts and Telecommunication: Chongqing, China, 2024.
17. Elgamily, M.K.; Mohamed, A.M.; Taleb, A.M.A.; Ata, M.M. Enhanced object detection in remote sensing images by applying metaheuristic and hybrid metaheuristic optimizers to YOLOv7 and YOLOv8. *Sci. Rep.* **2025**, *15*, 7226. [CrossRef]
18. Pan, H.; Jiao, W.; Yan, T.; Rehman, A.U.; Wan, A.; Yang, S. Combining kernel principal component analysis and spatial group-wise enhance convolutional neural network for fault recognition of rolling element bearings. *Meas. Sci. Technol.* **2023**, *34*, 125003. [CrossRef]
19. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A lightweight-design for real-time detector architectures. *J. Real-Time Image Process.* **2024**, *21*, 62. [CrossRef]
20. Shengde, C.; Junyu, L.; Xiaojie, X.; Jianzhou, G.; Shiyun, H.; Zhiyan, Z.; Yubin, L. Detection and tracking of agricultural spray droplets using GSConv-enhanced YOLOv5s and DeepSORT. *Comput. Electron. Agric.* **2025**, *235*, 110353. [CrossRef]
21. Cai, S.; Zhang, X.; Mo, Y. A Lightweight underwater detector enhanced by Attention mechanism, GSConv and WIoU on YOLOv8. *Sci. Rep.* **2024**, *14*, 25797. [CrossRef]
22. Guo, Y.; Liu, M.; Zeng, K.; Chang, Y. A novel farmland pest identification method based on transfer learning and improved YOLOv7. *Orient. Insects* **2025**, *59*, 165–183. [CrossRef]

23. Zhao, J.; Shen, C.; Huang, M.; Qi, Y.; Chai, Y.; Zheng, S. Deep learning accelerated micrograph-based porosity defect quantification in additively manufactured steels for uncovering a generic process-defect-properties relation. *Mater. Charact.* **2025**, *225*, 115094. [CrossRef]
24. Qiu, Z.; Rivaz, H. Is visual explanation with Grad-CAM more reliable for deeper neural networks? A case study with automatic pneumothorax diagnosis. In *Machine Learning in Medical Imaging*; Springer Nature: Cham, Switzerland, 2023; pp. 224–233.
25. Lv, Z.; Li, Y.; Qian, S.; Wu, L. Semi-supervised surface defect segmentation of aluminum strips based on self-attention consistency and cross-view pseudo supervision. *Adv. Eng. Inform.* **2025**, *66*, 103389. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.