metals

MDPI

Article

SDD-YOLO: A Lightweight, High-Generalization Methodology for Real-Time Detection of Strip Surface Defects

Yueyang Wu [1,†], Ruihan Chen [1,2,†], Zhi Li [1,†], Minhua Ye [3] and Ming Dai [1,*]

[1] School of Mathematics and Computer, Guangdong Ocean University, Zhanjiang 524008, China; aka@stu.gdou.edu.cn (Y.W.); 202111621104@stu.gdou.edu.cn (R.C.)
[2] Artificial Intelligence Research Institute, International (Macau) Institute of Academic Research, Macau 999078, China
[3] College of Ocean Engineering and Energy, Guangdong Ocean University, Zhanjiang 524008, China
* Correspondence: daiming@gdou.edu.cn
† These authors contributed equally to this work and share the first authorship.

Abstract: Flat-rolled steel sheets are one of the major products of the metal industry. Strip steel's production quality is crucial for the economic and safety aspects of humanity. Addressing the challenges of identifying the surface defects of strip steel in real production environments and low detection efficiency, this study presents an approach for strip defect detection based on YOLOv5s, termed SDD-YOLO. Initially, this study designs the Convolution-GhostNet Hybrid module (CGH) and Multi-Convolution Feature Fusion block (MCFF), effectively reducing computational complexity and enhancing feature extraction efficiency. Subsequently, CARAFE is employed to replace bilinear interpolation upsampling to improve image feature utilization; finally, the Bidirectional Feature Pyramid Network (BiFPN) is introduced to enhance the model's adaptability to targets of different scales. Experimental results demonstrate that, compared to the baseline YOLOv5s, this method achieves a 6.3% increase in mAP_{50}, reaching 76.1% on the Northeastern University Surface Defect Database for Detection (NEU-DET), with parameters and FLOPs of only 3.4MB and 6.4G, respectively, and FPS reaching 121, effectively identifying six types of defects such as Crazing and Inclusion. Furthermore, under the conditions of strong exposure, insufficient brightness, and the addition of Gaussian noise, the model's mAP_{50} still exceeds 70%, demonstrating the model's strong robustness. In conclusion, the proposed SDD-YOLO in this study features high accuracy, efficiency, and lightweight characteristics, making it applicable in actual production to enhance strip steel production quality and efficiency.

Keywords: strip defect detection; SDD-YOLO; YOLOv5s; BiFPN; lightweight; NEU-DET

check for updates

Citation: Wu, Y.; Chen, R.; Li, Z.; Ye, M.; Dai, M. SDD-YOLO: A Lightweight, High-Generalization Methodology for Real-Time Detection of Strip Surface Defects. *Metals* **2024**, *14*, 650. https://doi.org/10.3390/met14060650

Academic Editor: Olivier Pantale

Received: 5 May 2024
Revised: 26 May 2024
Accepted: 27 May 2024
Published: 30 May 2024

Copyright: © 2024 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 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Strip steel, as a crucial core product in the metal industry, plays an indispensable role in various fields such as construction, automotives, machinery manufacturing, aerospace, and beyond [1]. With the flourishing development of high-end industries like precision metal manufacturing, the metal industry has increasingly stringent requirements for the quality of strip steel products. Throughout the production process of steel materials, various factors such as raw material quality, manufacturing equipment, and the production environment influence the occurrence of diverse types of defects on the product surface, including cracks, voids, and scratches, among others [1]. These defects cause significant economic losses to the metal manufacturing industry and pose serious safety hazards to society. In recent years, the field of metal surface defect detection has garnered increasing attention, with notable improvements in the effectiveness and efficiency of detection technology [2]. However, the detection of metal surface defects is prone to interference from various factors during the production process, such as light reflection, variations in light intensity, and material properties, thereby augmenting the challenge of metal surface defect detection [3].

Therefore, enhancing the capability of strip steel surface defect detection is of paramount importance for improving product quality and manufacturing efficiency [4].

Since the late 20th century, scholars in the metal industry have been dedicated to researching the detection and classification of metal surface defects. Initially, detection methods primarily comprised eddy current testing [5], infrared detection [6], magnetic particle testing [7], and visual inspection; however, these methods are costly and inefficient. In recent years, with the continuous advancement of computer vision technology, object detection-based metal surface defect detection technology has been widely applied in industrial production and gradually replaced traditional detection methods [8,9].

As a significant research direction in computer vision, object detection can be categorized into two types based on feature extraction methods: traditional object detection methods and deep learning methods. Traditional object detection methods can be broadly divided into three categories. Firstly, methods such as Local Binary Patterns (LBP) [10], Histogram of Oriented Gradients (HOG) [11], and Gray Level Co-occurrence Matrix (GLCM) [12] extract the features of surface defects by manually designing parameters [13]. The second category comprises techniques based on statistical and spectral methods, such as Fourier Transform [14], Wavelet Transform [15], and Gabor Filters [16]. The last category comprises methods based on machine learning models, such as autoregressive models [17] and Markov Random Field models [18]. While these methods have made certain advancements in the field of metal surface defect detection, they are limited by the sensitivity of images to lighting conditions and backgrounds, and the inability of shallowly extracted manually designed features to effectively represent images with complex backgrounds. Therefore, despite the development of various traditional machine learning-based metal surface defect detection models, these models still fail to be effectively applied in practical production [19].

With the continuous development of artificial intelligence technology and the improvement of GPU performance, deep learning technology has shown unique application potential in metal surface defect detection [20]. Convolutional Neural Networks (CNNs) have been highly acclaimed for their powerful feature extraction capabilities, and many scholars have applied deep learning technology to the research of metal surface defect detection [21]. For instance, Lin et al. [22] proposed a multi-model cascaded CNN based on MobileNet, aiming to reduce false positives in industrial optical defect detection without considering detection speed. Li et al. [23] developed a parameter-complex integrated framework for industrial railway defect detection, aiming to improve the detection performance for railway defects. Zhou et al. [24] combined attention mechanism modules with the YOLOv5s model, improving detection performance while reducing detection efficiency. Zhang et al. [25] combined the lightweight convolutional layer GSConv with YOLOv5s to increase the detection rate of strip steel surface defects at the cost of reducing detection accuracy. Lv et al. [26] proposed a high-precision strip steel surface defect detection model based on the improved YOLOv7. Li et al. [27] improved on YOLOv7, maintaining high defect detection capabilities while slightly reducing model complexity. Although these studies have made certain contributions to the field of metal surface defect detection, these models have not yet achieved a good balance between detection accuracy and efficiency. Furthermore, since the performance of these detection models is mainly evaluated on ideal environment datasets, the detection of metal surface defects in practical applications, especially strip steel surface defect detection, is influenced by factors such as overexposure and uneven brightness. Therefore, current metal surface defect detection models still cannot overcome these challenges and achieve both detection accuracy and speed in practical strip steel surface defect detection applications.

To address the current issues in strip steel surface defect detection applications, this study proposes a lightweight, highly generalized real-time strip defect detection method named SDD-YOLO. While maintaining excellent detection performance and efficiency, this method features simpler parameters and a lighter model, meeting the demands of the metal forging industry. The main contributions of this study are as follows:

(1) The proposal of the CGH module, which improves the C3 module of YOLOv5. GhostConv is used to replace conventional convolution layers on bottleneck layers and branches, effectively reducing computational complexity. Meanwhile, residual connections are introduced to replace Concat operations, significantly enhancing the stability of the network. This module combines the low-cost feature map generation of GhostConv with the characteristics of residual connections to reduce memory consumption and improve feature extraction efficiency.
(2) The proposal of the MCFF module. By employing convolution kernels of different sizes and channel attention mechanisms, the feature expression capability and robustness of the model are effectively enhanced, avoiding gradient disappearance, accelerating the training process, and ensuring that the neural network can accomplish more complex tasks.
(3) By simultaneously introducing CARAFE and BiFPNs, the receptive field of convolutional neural networks is enhanced, and the resolution of feature maps is improved, resulting in a more effective upsampling process within convolutional neural networks. Furthermore, the incorporation of the BiFPN into the Neck layer of YOLOv5, in place of traditional FPN and PANet structures, enhances the model's capability for deep feature fusion.
(4) This study considers disruptive factors in strip steel production and uses three data interference methods—high exposure, low brightness, and Gaussian noise—on the original data to test the robustness of the SDD-YOLO method. The results show that SDD-YOLO has advanced generalization performance, making it suitable for real production environments and effectively improving the quality and efficiency of strip steel production.

The remainder of this study is organized as follows: Section 2 introduces related research on the original YOLOv5s network, multiscale feature fusion, and lightweight networks. Then, Section 3 describes the dataset and the proposed SDD-YOLO method. The detailed analysis of the experimental process is presented in Section 4. Finally, Section 5 summarizes the work of this study.

2. Related Work

2.1. YOLOv5

The You Only Look Once (YOLO) series is regarded as one of the classics in object detection technology [28]. Its fifth generation (YOLOv5) was introduced in 2020 [29] and is considered one of the cutting-edge object detection algorithms in the field of deep learning. The implementation method mainly involves dividing the entire image into a series of grids and predicting various information for each grid, including the presence of objects, their positions, sizes, categories, etc. YOLOv5 has been thoroughly tested on several common deep learning techniques, selecting effective techniques to achieve satisfactory experimental results. On Tesla V100, YOLOv5 achieves real-time detection speeds of 156 FPS on the COCO2017 dataset with an accuracy of 56.8% AP. In recent years, YOLOv5 has been widely applied in various fields such as industry [30,31], agriculture [32,33], etc. The structure of YOLOv5 mainly consists of four parts. The first part is the Input, including image data augmentation concatenation, setting three initial anchors, and adaptive scaling of image size. The structures of the remaining three parts are illustrated in Figure 1.

Backbone: The backbone network of YOLOv5 consists of three parts: CBS, C3, and SPPF, which convert the original image into multi-layer feature maps and extract key features for subsequent object detection. The CBS module is the cornerstone of convolutional neural networks, encompassing Conv [34], BatchNorm [35], and SiLU [36], to extract local spatial information from images and endow them with the magic of nonlinear transformation. The C3 module, focusing on high accuracy, ingeniously enhances the computational efficiency of the network, enabling a higher level of speed and efficiency for object detection. For feature extraction, Cross-Stage Partial Networks (CSP) [37] and Spatial Pyramid Pooling Fusion (SPPF) [38] are employed to extract feature maps of different sizes

from input images. The clever design of CSP not only reduces computational burden but also improves inference speed. Meanwhile, the SPPF module, like a spatial pyramid, can handle images of different resolutions, and expand the perception range while reducing the computational burden, thus comprehensively refining the overall features of the targets.

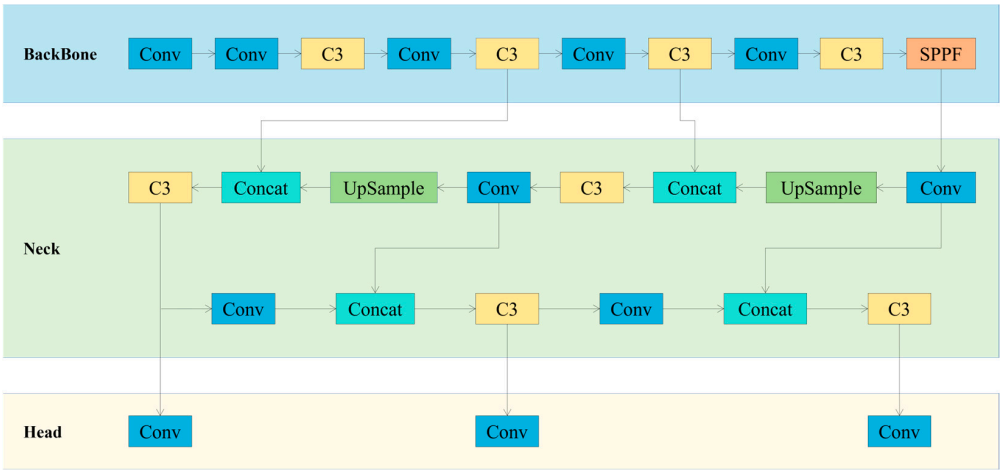

Figure 1. The network architecture diagram of YOLOv5.

Neck: The Neck of YOLOv5 adopts a fusion structure of FPNs [39] and PANs [40], combining traditional FPN layers with bottom-up Feature Pyramid Networks (PAN), and cleverly integrating extracted semantic features with positional features. Simultaneously, the fusion of backbone network layers with detection layers injects richer feature information into the model. These two structures complement each other, enhancing features extracted from different network layers, further improving detection accuracy and capability.

Head: The Head is mainly used to predict targets of different sizes on feature maps. YOLOv5 inherits the multiscale prediction Head of YOLOv4 and integrates three layers of feature mapping to enhance the detection performance of targets of different sizes. The Head of YOLOv5 employs three detection Heads responsible for detecting target objects and predicting their categories and positions. These three Heads correspond to feature maps of 20×20, 40×40, and 80×80, accurately outputting targets of different sizes [41].

In the YOLOv5 model series, there are four models, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, which are divided according to different depth and width parameters. With the increase in model type, the performance gradually improves, but it is accompanied by the complexity of network structure and the slowing down of detection speed. However, in this study, the steel strip surface defect detection model needs to display surface defects in real time and minimize the consumption of operating memory. Therefore, this study selected the relatively simple network structure and fast detection speed of YOLOv5s as the baseline model.

2.2. Multi-Scale Feature Fusion

Multi-scale feature fusion is a powerful tool for improving model performance, especially in the field of object detection [42]. Traditional neural networks often use fixed-size filters or pooling operations to process input images, which often leads to the loss of low-level details or high-level semantic information. To address this issue, introducing multiscale feature fusion becomes an inevitable choice. There are various methods to achieve multiscale feature fusion, among which a common approach is to concatenate or overlay feature maps of different scales, enabling the network to integrate information from various scales for decision-making. Another approach is to generate feature maps of different levels using a pyramid structure and then fuse them to capture details and semantic information at different scales. Through multiscale feature fusion, the model can better adapt to objects or scenes of different scales and sizes, enhancing model robustness

and performance in complex scenarios. However, traditional top-down Feature Pyramid Networks (FPNs) often fail to fully utilize features of different scales due to the limitation of unidirectional information flow, thus requiring more effective methods to address this issue [43]. The YOLOv5 algorithm adopts the Path Aggregation Network (PANet) network for feature fusion, which introduces a bottom-up Path Aggregation Network compared to FPNs, realizing bidirectional information flow [44]. However, the PANet network requires more parameters and computational resources, resulting in slower speeds, making it unsuitable for real-time object detection. If low-level feature information is insufficient or some information is lost, the PANet method may lead to decreased detection accuracy. The BiFPN is proposed as a novel network structure for multiscale feature fusion. Compared to traditional unidirectional FPNs, the BiFPN improves fusion accuracy and efficiency by utilizing bidirectional connections and feature node fusion in the Feature Pyramid Network, solving the problem of unidirectional FPNs' inability to fully utilize feature information at different scales [45–47]. Therefore, in this study, the original FPN and PANet structures in YOLOv5 are improved to the BiFPN network to achieve more efficient multiscale feature fusion.

2.3. Lightweight Network

The lightweight network is a neural network model designed for scenarios with limited computational resources. Its design aims to maintain good performance while minimizing the number of model parameters and computational complexity as much as possible. Typically, lightweight networks adopt various optimization strategies such as simplifying network structures, reducing parameter count, and lowering network layer complexity to efficiently operate in resource-constrained environments. This type of network has wide applications in scenarios such as mobile devices, embedded systems, and edge computing, meeting the demands of limited computational and storage resources while achieving fast, accurate inference and processing tasks.

To reduce computational costs while maintaining model detection efficiency, researchers have proposed various methods. Some methods focus on reducing the precision of weights to make the model more compact [48]. Additionally, there are a series of methods aimed at reducing the number of less important parameters in pruned training models. For example, MADNet [49] is a dense lightweight network designed to achieve stronger multiscale feature expression and feature correlation learning. In terms of feature extraction, Liu et al. [50] constructed a network with expanded convolutions and attention modules, using pooling operations of different sizes to encode surrounding semantic information.

However, these methods typically achieve compression of pre-trained networks or direct training of small-scale networks, rather than solely focusing on model size while ignoring overall performance. Taking into account its performance comprehensively, the SDD-YOLO network proposed in this study effectively reduces computational complexity and model size, truly achieving lightweight and efficient characteristics.

3. Materials and Methods

3.1. Dataset

To validate the effectiveness of the proposed model, this study selected the Northeastern University Surface Defect Database for Detection (NEU-DET) [51] to evaluate the performance of SDD-YOLO and other models. The NEU-DET dataset consists of six types of defects: Crazing (Cr), Inclusions (In), Patches (Pa), Pitted Surface (Ps), Rolled-in Scale (Rs), and Scratches (Sc). Each type of defect contains 300 grayscale images with a resolution of 200 × 200 pixels, totaling 1800 images. In this study, the NEU-DET dataset was divided into training, validation, and test sets in proportions of 80%, 10%, and 10%, respectively. The training set was used to optimize network parameters to minimize the loss function, the validation set was used to validate the performance of the model during training, and the test set was used to evaluate the accuracy of the trained network in surface defect recognition. Figure 2 shows samples of the six typical surface defects in NEU-DET.

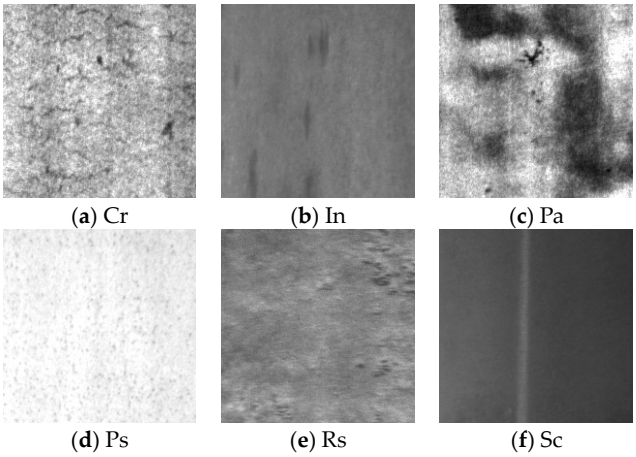

Figure 2. Six sample images from the NEU-DET dataset.

3.2. Methods

YOLOv5s, recognized as a lightweight neural network for object detection, exhibits relatively lower costs in inference computation and training speed. To strike a balance between detection speed and accuracy, this study opted to utilize YOLOv5s as the foundation for improving the identification network in this investigation. Building upon the YOLOv5 network with C3 as its backbone, which maintains a relatively fast speed while enhancing detection performance, this study introduces the CGH module based on the C3 structure and establishes the novel network SDD-YOLO. Furthermore, this study introduces a novel feature fusion method termed MCFF. This method employs convolutions with larger receptive fields to extract richer feature scales and adaptively extract features, thereby augmenting the network's multiscale recognition capability for surface defects. To enhance the receptive field of convolutional neural networks and the resolution of feature maps, CARAFE replaces traditional upsampling methods in this study. Finally, the incorporation of BiFPNs enables SDD-YOLO to combine more feature information while conserving computational resources, thereby enhancing the network's information extraction capability.

Building upon the YOLOv5s network structure, this study introduces CARAFE and BiFPNs, combined with the CGH module and MCFF module, proposing an SDD-YOLO network for strip steel surface defect detection. The network structure is illustrated in Figure 3. Subsequently, this study provides a detailed explanation of the CGH module and MCFF module proposed in this study.

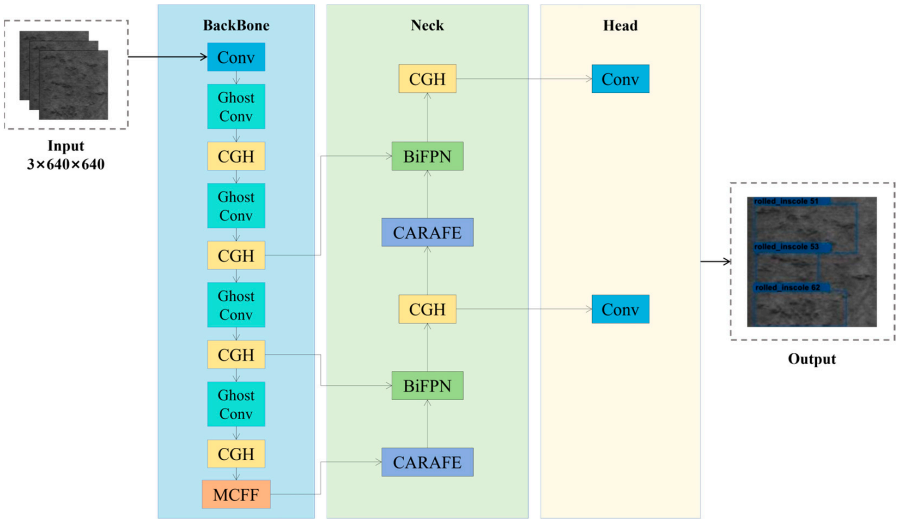

Figure 3. The network architecture diagram of SDD-YOLO.

3.2.1. CGH Module

YOLOv5 is a classic object detection model, with its C3 module structure comprising three standard convolutional layers and a bottleneck layer. Although the C3 module adopts the CSP (Cross Stage Partial) structure to reduce computational complexity, its complexity remains relatively high compared to traditional single convolutional structures. This may increase the time cost of training and inference. To address this issue, this study proposes the CGH module. The CGH module replaces the bottleneck layer with GhostConv based on C3 and also replaces the conventional convolutional layers on the branches with GhostConv. Additionally, to address issues such as overfitting, gradient vanishing, and gradient explosion caused by excessive network depth, this study utilizes residual connections to replace Concat operations and removes the last conventional convolutional layer. The structure of the CGH module is illustrated in Figure 4.

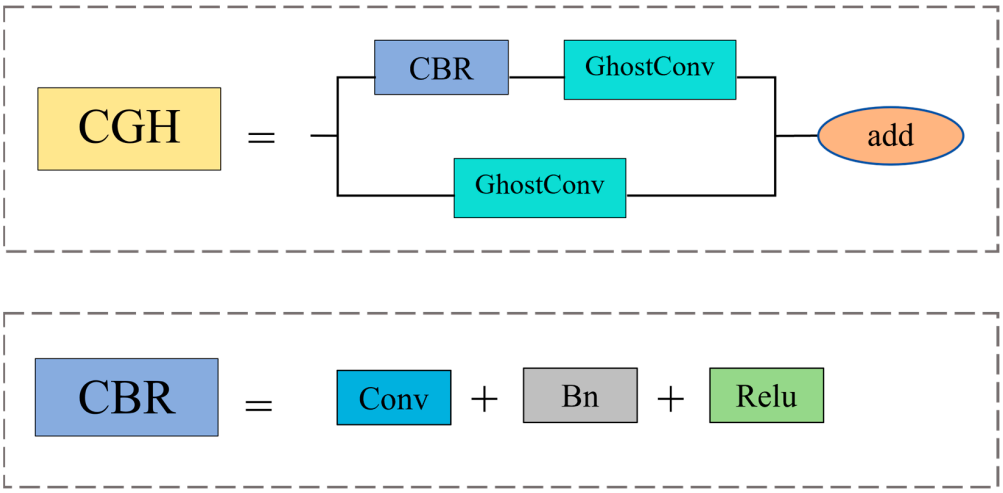

Figure 4. The structural diagram of the CGH module.

Deep neural networks often generate many similar redundant feature maps during feature extraction, consuming a large amount of computational resources. Although these feature maps are crucial for network understanding of data features, their generation process is costly. Inspired by GhostNet [52], which was designed to validate the effectiveness of GhostConv, this study introduces the GhostConv technique to reduce memory consumption during feature space expansion. GhostConv generates more feature maps in a lower-cost manner, thereby reducing memory consumption during intermediate expansion. Additionally, to ensure effective feature extraction and enhance network stability, this study introduces residual connections in the CGH module. The structure of GhostConv is depicted in Figure 5.

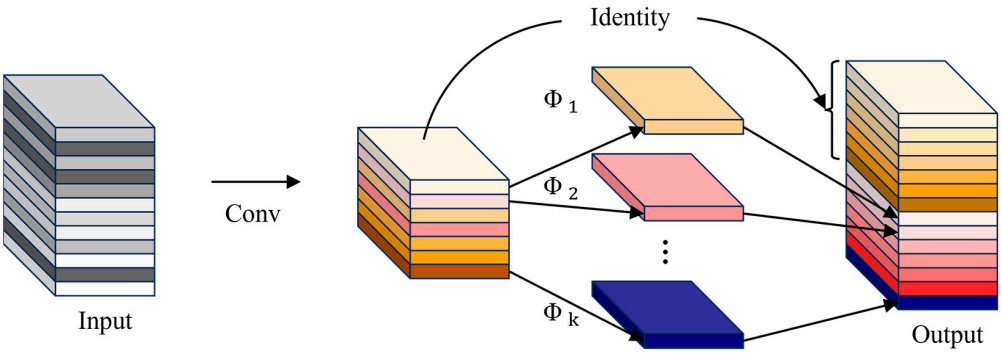

Figure 5. The structural diagram of GhostConv.

Residual connections address problems that may arise from increasing network depth, such as overfitting, gradient vanishing, and gradient explosion. In this study, residual connections are utilized in the CGH module to mitigate overfitting issues, significantly improving network stability. The Equation (1) for computing residual connections is as follows:

$$Output = F(Input, \{W_i\}) + Input, \tag{1}$$

The input and output of the first layer are defined as Input and Output, respectively, with nonlinear transformation defined as $F(Input, \{W_i\})$, including nonlinear activation functions, etc. The introduction of the GhostConv module and residual structure in the CGH module can greatly reduce computational complexity, obtain sufficient feature maps, and ensure network stability.

The CGH module improves the YOLOv5 C3 module by replacing the bottleneck layer and conventional convolutional layers on the branches with GhostConv, effectively reducing computational complexity. Furthermore, the introduction of residual connections instead of Concat operations addresses issues such as overfitting, gradient vanishing, and gradient explosion that deep networks may encounter, significantly enhancing network stability. This module combines the characteristics of low-cost feature map generation with GhostConv and residual connections to reduce memory consumption and improve feature extraction efficiency.

3.2.2. Multi-Convolution Features Fusion Block

For enhanced multiscale feature extraction, this study introduces a Multi-scale Context Fusion (MCFF) block. The schematic diagram of the MCFF block is illustrated in Figure 6. M_{in} represents the input of the MCFF block, from which feature maps M_1, M_2, and M_3 are extracted using 3×3, 5×5, and 7×7 convolutional kernels of $M_{in} \in R^{C \times H \times W}$, respectively. Leveraging convolutions with larger receptive fields enables the extraction of richer feature scales. Additionally, the proposed block employs Global Average Pooling (GAP) to extract features from different resolutions of M_2 and M_3 Subsequently, adaptive feature extraction is performed using one-dimensional convolution. Through Sigmoid transformation, channel attention $S_2 \in R^{1 \times 1 \times C}$ and $S_3 \in R^{1 \times 1 \times C}$ are obtained and utilized in conjunction with function CBR to fuse M_1, S_2, and S_3, resulting in the final output feature $M_{out} \in R^{C \times H \times W}$, as depicted in Equation (2).

$$\begin{aligned} M_{out} &= CBR(M_1 \otimes S_2 \otimes S_3) \\ &= \sigma(BatchNorm(Conv^{3\times3}(M_1 \otimes S_2 \otimes S_3))), \end{aligned} \tag{2}$$

where CBR represents the combination of three layers: a 3×3 convolutional layer, a Batch Normalization layer, and a non-linear activation function ReLU. $\otimes$ denotes element-wise multiplication, σ signifies the ReLU layer, $Conv^{3\times3}$ is a 3×3 convolutional layer used for fusing feature maps of different resolutions and channel attention, mitigating feature misalignment issues caused by simple multiplication. *BatchNorm* is a normalization technique addressing inconsistent input data distributions, highlighting their relative differences, thereby accelerating training speed. The ReLU layer introduces non-linear relationships to feature layers, preventing gradient vanishing and overfitting, and ensuring the network's capability to accomplish complex detection tasks.

In the domain of metal surface defect detection, the application of MCFF blocks offers several advantages. Firstly, the MCFF block is a crucial component designed specifically for multiscale feature extraction. By employing convolutional kernels of varying sizes and channel attention mechanisms, the MCFF block effectively enhances the model's feature expression capability and robustness. This implies that the model can better capture various scales and shapes of metal surface defects, thereby improving the accuracy and comprehensiveness of defect detection. Secondly, the introduction of MCFF blocks aids in mitigating gradient vanishing issues and accelerates the model's training process. Metal surface defect detection tasks often entail handling large volumes of data and complex

features. Through optimized feature extraction, MCFF blocks enable neural networks to learn and adapt to different surface defect patterns more efficiently, thereby enhancing the model's convergence speed and training efficiency. Most importantly, MCFF blocks ensure that neural networks can tackle complex strip steel surface defect detection tasks. Surface defects on metal surfaces may exhibit different shapes, sizes, and textures, thus requiring models with strong feature expression and generalization capabilities. The introduction of MCFF blocks enables the model to better understand and distinguish between different types of defects, thereby enhancing the robustness and reliability of the detection system.

In summary, the application of MCFF blocks in strip steel surface defect detection tasks effectively enhances the model's performance and strengthens its ability to detect defects of different scales and shapes, while also improving training efficiency and generalization capability. This provides the metal manufacturing industry with a more reliable and efficient quality control solution.

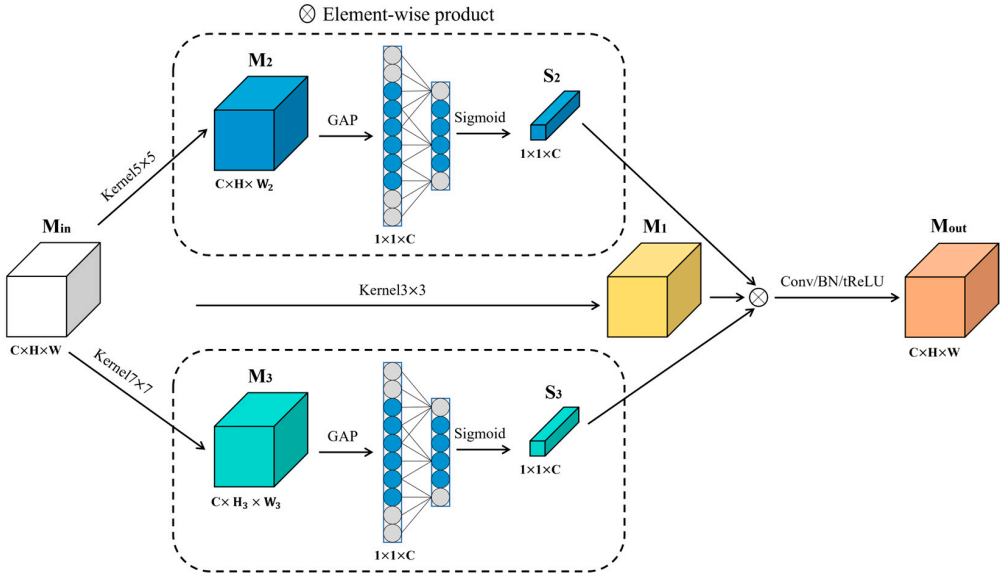

Figure 6. The structural diagram of multi-convolution features fusion block (MCFF).

3.2.3. CARAFE

Feature upsampling is an essential operation in image processing and a key operation in modern convolutional network architectures, used to convert low-resolution feature maps into high-resolution ones, thereby enhancing the model's ability to capture details and local information. Currently, there are two mainstream upsampling methods. One is linear interpolation, including nearest neighbor interpolation and bilinear interpolation, widely used in sub-pixel space but unable to fully capture semantic information, which may lead to feature loss. The other common method is deconvolution, which expands the size through convolutional operations. However, deconvolution typically uses the same convolution kernel to operate on the entire feature map, limiting its perception of local variations, making it difficult to effectively capture local details, and increasing the model's parameter count.

Wang and his team proposed the CARAFE upsampling operator [53], which introduces the Content-Aware ReAssembly in FEature space (CARAFE) technology into feature map sampling. The CARAFE upsampler utilizes content information at each position to predict the reassembled kernel and reassemble features within a predefined neighborhood. Compared to traditional methods, the CARAFE upsampler achieves significant progress with only a small number of additional parameters and amount of computational work. Since CARAFE can flexibly adjust and optimize the reassembled kernel based on content information at different positions, it outperforms mainstream upsampling operators such as

interpolation or deconvolution in terms of performance. The network structure of CARAFE is depicted in Figure 7.

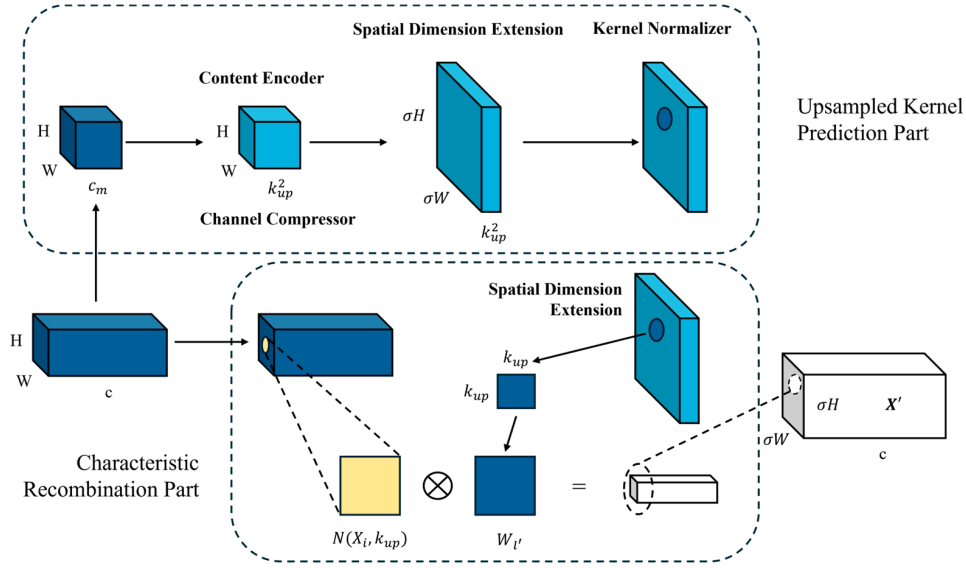

Figure 7. The structural diagram of the CARAEF upsampling operator.

In the context of metal surface defect detection, the application of CARAFE technology brings significant benefits. Firstly, by replacing traditional upsampling methods, CARAFE enhances the receptive field and resolution of convolutional neural networks. This means the network can better capture subtle features and details in images, thereby improving the ability to identify defects on metal surfaces. Secondly, CARAFE technology makes the upsampling process in convolutional neural networks more effective. Traditional upsampling methods may introduce blur or distortion, especially for metal surface images containing a large amount of detailed information. CARAFE can more accurately reconstruct feature maps to better adapt to the content of input images. As a result, the detected defect positions and shapes are more accurate, enhancing the robustness and accuracy of the detection system.

In summary, the application of CARAFE technology in metal surface defect detection can effectively improve the performance of the detection system and enhance the ability to identify defects, while maintaining the clarity and accuracy of image features, thus providing strong support for quality control in practical production.

3.2.4. BiFPN

In the YOLOv5 algorithm, a Feature Pyramid Network (FPN) combined with a Path Aggregation Network (PAN) structure is employed for Neck region processing, achieving favorable outcomes in multiscale fusion. However, due to its computational complexity and the susceptibility of task images to environmental factors alongside diverse scales, there exists insufficient extraction and utilization of structural features, consequently resulting in substantial loss errors. To address this issue, a model named the Weighted Bi-directional Feature Pyramid Network (BiFPN), proposed by Google's artificial intelligence research team including Mingxing Tan et al., is introduced [54]. This innovation allows rapid and straightforward multiscale feature fusion. The BiFPN module employs a weighted feature fusion mechanism to learn the importance of different resolution feature information in input images, as demonstrated in Equations (3) and (4). Simultaneously, it adopts a fast normalization method, as illustrated in Equation (5). Consequently, the BiFPN structure is

integrated into the Neck region of the network. The structures of FPNs, PANs, and BiFPNs are depicted in Figure 8.

$$P_i^{td} = Conv\left(\frac{\omega_1 \cdot p_i^{in} + \omega_2 \cdot resize(P_{i+1}^{in})}{\omega_1 + \omega_2 + \varepsilon}\right), \tag{3}$$

$$P_i^{out} = Conv\left(\frac{\omega'_1 \cdot p_i^{in} + \omega'_2 \cdot p_i^{td} + \omega'_3 \cdot resize(P_{i-1}^{out})}{\omega'_1 + \omega'_2 + \omega'_3 + \varepsilon}\right), \tag{4}$$

In Equations (3) and (4), P_i^{in} represents the input sample feature information of the i th layer node, P_i^{td} denotes the intermediate feature information of the top-down transmission path of the i th layer, P_i^{out} signifies the output feature information of the bottom-up transmission path of the i th layer, *Conv* indicates convolution operation, and *resize* represents either upsampling or downsampling operation.

$$O = \sum_i \frac{\omega_i \cdot I_i}{\varepsilon + \sum_j \omega_j}, \tag{5}$$

In Equation (5), O represents the output value, I_i denotes the input value of the node, j signifies the summation of all input nodes, and ω_i represents the weight of input nodes. To ensure the condition of each input node's weight $\omega_i \geq 0$ holds, the Rectified Linear Unit RELU activation function is applied to each operation.

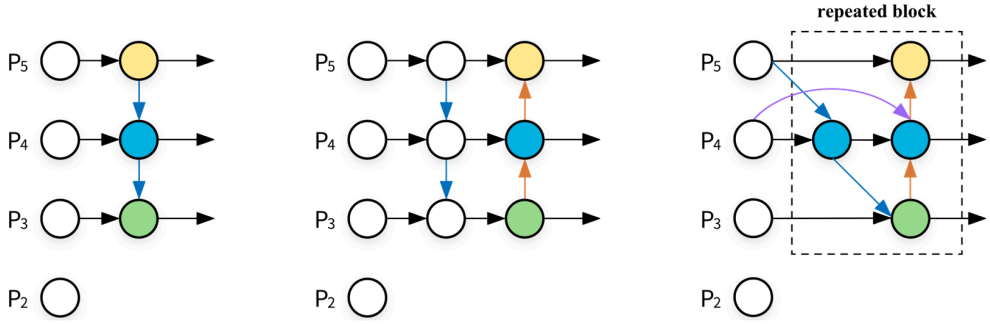

Figure 8. The structural diagrams of FPNs, PANs, and BiFPNs in YOLOv5s.

The BiFPN is constructed based on PANs. Compared to the original Neck structure, the BiFPN removes nodes that do not contribute to feature fusion to save resources. It introduces new channels between input and output nodes at the same level to more fully integrate feature information. Simultaneously, a cross-scale connection method is proposed, and additional edges are added to directly fuse features in the feature extraction network with features of relative size in the bottom-up path, retaining more surface-level semantic information while minimizing the loss of deep semantic information. The introduction of BiFPNs enables SDD-YOLO to save computational resources while incorporating more feature information, enhancing the network's information extraction capabilities. It combines bottom-layer position information with high-level semantic information, further improving the network's performance in object detection tasks.

3.3. Experimental Parameter Settings

In this experiment, an NVIDIA GeForce RTX 3090 graphics card with 24GB of memory and an Intel(R) Xeon(R) Silver 4210 @ 2.20GHz CPU with 32GB of memory were utilized. The experiment was conducted using the PyTorch deep learning framework on a Windows 10 environment for both training and testing. The training process of the network consisted of 160 epochs. This study employed the stochastic gradient descent (SGD) optimizer with a batch size of 8 and employed a linearly decaying learning rate scheduling strategy, with an initial learning rate set to 0.01 and a final learning rate of 0.0001. The momentum parameter was set to 0.941, and the weight decay was set to 0.0005. Input images were uniformly

resized to 640 × 640 dimensions and normalized. Specific training parameter settings are outlined in Table 1.

Table 1. Training parameters.

Parameter Name	Parameter Value
Initial learning rate	0.01
Momentum	0.941
Weight decay	0.0001
Epochs	160
Batch size	8
Noautoanchor	FALSE
Input image size	640
Optimizer	SGD

3.4. Model Evaluation Metrics

This study comprehensively evaluates the proposed network using metrics such as accuracy (AP), mean average precision (mAP), precision, recall, floating-point operations (FLOPs), parameter count (Params), and frames per second (FPS). In the task of strip steel surface defect detection, Intersection over Union (IOU) is employed to determine whether the detection result corresponds to a genuine defect. If this value exceeds a predefined threshold, it is considered a positive sample; otherwise, it is deemed a negative sample. In object detection tasks, precision and recall are crucial metrics for evaluating the recognition performance of the network.

Precision (P) is defined as the ratio of the number of correctly classified positive samples to the total number of samples classified as positive by the classifier (Equation (6)).

$$Precision = \frac{TP}{TP + FP}, \tag{6}$$

Recall (R) is defined as the ratio of the number of correctly classified positive samples to the total number of true positive samples (Equation (7)).

$$Recall = \frac{TP}{TP + FN}, \tag{7}$$

In Equations (6) and (7), TP denotes the number of samples correctly predicted as positive by the model, FP denotes the number of samples incorrectly predicted as positive by the model, and FN denotes the number of samples incorrectly predicted as negative by the model.

AP is a metric that summarizes the precision–recall curve and measures the precision at various recall levels for a specific class. mAP is the mean of average precision scores for all classes, providing a single metric that evaluates the overall performance of a model across multiple classes. Their formulas are shown as Equations (8) and (9).

$$AP = \int_0^1 P(R)dR, \tag{8}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n}, \tag{9}$$

In Equation (9), mAP is the approximate area enclosed by the precision–recall curve.

FLOPs are commonly used to measure model complexity, with lower values indicating faster model execution rates, as shown in Equations (10) and (11).

$$FLOPs(Conv) = (2 \times C_{in} \times K^2 - 1) \times W_{out} \times H_{out} \times C_{out}, \tag{10}$$

$$FLOPs(Conv) = (2 \times C_{in} - 1) \times C_{out}, \tag{11}$$

In Equations (10) and (11), C_{in} and C_{out} represent input and output channels, respectively, and K, H_{out}, and W_{out} represent kernel size, output feature map height, and width, respectively.

In evaluating the performance of the models in this study, $mAP_{50:95}$ and mAP_{50} were simultaneously employed. $mAP_{50:95}$ represents the average mAP (mean average precision) across different IOU thresholds (ranging from 0.5 to 0.95 with a step size of 0.05), providing a comprehensive reflection of the model's performance. Additionally, during the testing phase, FPS was used to indicate the inference speed, with results averaged over 180 test images. To compare the computational complexity of different networks, this study selected computational time complexity (FLOPs) and computational space complexity (Params, parameter count) to represent the differences between various methods.

4. Experimental Result and Discussion

To demonstrate the excellent performance of SDD-YOLO in strip steel surface defect detection, this section presents the experimental results and analysis. In this section, this study first compares the performance of SDD-YOLO with the baseline YOLOv5s. Subsequently, ablation experiments are conducted to validate the contributions of the CGH, GhostConv, MCFF, CARAFE, and BiFPN modules, and the specific contributions of the Feature Pyramid Networks FPN, NAS-FPN, and BiFPN are verified through experiments. Additionally, the effectiveness of the proposed method is validated by comparing it with other classical object detection methods applied to strip steel surface defect detection tasks. In the final section of this study, three data augmentation methods—increasing brightness, decreasing brightness, and adding Gaussian noise—are employed to perform robustness analysis on the proposed SDD-YOLO method, demonstrating the model's strong generalization ability.

4.1. Performance Evaluation

The SDD-YOLO model was validated using the NEU-DET dataset. Experimental results, as depicted in Table 2, illustrate that the proposed SDD-YOLO model achieved improvements of 4.93%, 3.28%, 6.3%, and 4.3% in accuracy, recall, mAP_{50}, and $mAP_{50:95}$, respectively, while reducing parameter count by 51.4%.

A comparison of the detection performance of six surface defect categories between the baseline YOLOv5s and SDD-YOLOv5 models is illustrated in Figure 9. It is evident that the SDD-YOLO proposed in this study not only handles various types and conditions of strip steel surface defect images but also exhibits significantly superior detection performance compared to YOLOv5s.

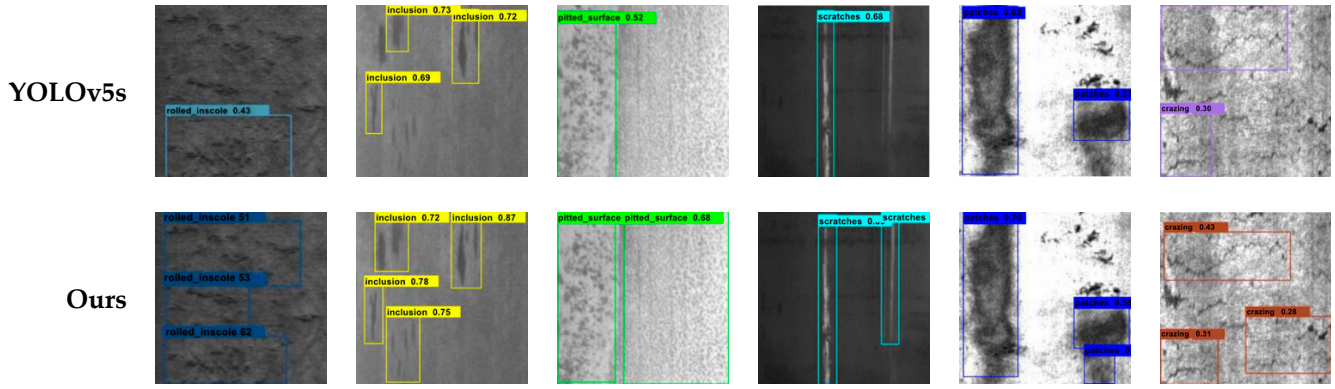

Figure 9. The comparison of defect detection results between YOLOv5s and SDD-YOLO on the NEU-DET dataset.

Table 2. Comparison of results between YOLOv5s model and SDD-YOLO model.

Model	P	R	mAP_{50} (%)	$mAP_{50:95}$ (%)	Params (M)
YOLOv5s	72.3	50.8	69.8	36.0	7.0
SDD-YOLO	78.6	53.9	76.1	40.3	3.4

4.2. Ablation Experiment

This study conducted ablation experiments to validate the advantages of the CGH, GhostConv, TCFF, CARAFE, and BiFPN modules in the SDD-YOLO network. The experimental results, as shown in Table 3, demonstrate that these modules can improve detection speed and accuracy, and reduce parameter count and computational complexity. However, there exists a trade-off among detection accuracy, parameter count, computational complexity, and detection speed among these modules. Experiment 16, which incorporates these five modules, achieved a 6.3% improvement in detection accuracy compared to Experiment 1 while reducing parameter count and computational complexity by 51% and 59%, respectively, and increasing detection speed by 25 frames per second. The results of Experiment 16 significantly outperformed other experiments. By effectively reducing computational complexity and model size while maintaining performance, this experiment achieved lightweight and efficient detection of strip steel surface defects. These findings suggest that, for real-time and accurate detection of strip steel surface defects, the combination of Experiment 16 is more suitable.

Table 3. Ablation experiments of SDD-YOLO.

Number	CGH	GhostConv	MCFF	CARAFE + BiFPN	mAP_{50}	$mAP_{50:95}$	Params (M)	FLOPs (G)	FPS
1	-	-	-	-	69.8	36.0	7.0	15.8	96
2	√	-	-	-	71.7	37.2	3.6	7.7	147
3	-	√	-	-	74.3	38.3	3.9	11.4	120
4	-	-	√	-	72.3	38.8	4.8	13.4	117
5	-	-	-	√	72.5	38.1	4.9	13.8	103
6	√	√	-	-	72.0	37.4	3.7	8.9	139
7	√	-	√	-	72.3	38.2	4.0	9.5	135
8	√	-	-	√	71.9	37.5	3.8	9.1	136
9	-	√	√	-	73.2	37.1	4.5	8.6	122
10	-	√	-	√	72.5	36.5	4.3	10.3	119
11	-	-	√	√	71.3	36.8	4.6	8.7	112
12	√	√	√	-	74.5	39.0	4.1	8.3	130
13	√	√	-	√	74.1	39.4	3.8	8.0	125
14	√	-	√	√	75.0	39.1	4.2	7.8	122
15	-	√	√	√	73.7	39.5	4.5	8.5	127
16	√	√	√	√	76.1	40.3	3.4	6.4	121

Currently, three commonly used Feature Pyramid Networks in the literature are FPNs, NAS-FPNs, and BiFPNs. As shown in Table 4, after integrating these three different Feature Pyramid Networks into SDD-YOLO, BiFPNs achieved the highest mAP_{50} and $mAP_{50:95}$ among the three, with this study emphasizing accuracy. Therefore, the BiFPN was chosen as the Feature Pyramid Network in SDD-YOLO.

Table 4. Comparison of different Feature Pyramid Networks in SDD-YOLO.

Number	FPN	NAS-FPN	BiFPN	mAP_{50}	$mAP_{50:95}$
1	√	-	-	70.9	37.1
2	-	√	-	72.3	38.2
3	-	-	√	76.1	40.3

4.3. Comparison of Different Modules

To validate the performance of the proposed SDD-YOLO method for strip steel surface defect detection, this study compared it with classical models, including YOLOv3, YOLOv5s, YOLOv7-tiny, and YOLOv8s, among others. Additionally, the default backbone network of YOLOv5s was replaced with lightweight backbone networks such as ShuffleNetv2, MobileNetv3, and GhostNet. Table 5 and Figure 10 present the comprehensive performance of each method on the NEU-DET dataset.

Table 5. Comparison of SDD-YOLO with state-of-the-art methods.

Method	mAP_{50}	$mAP_{50:95}$	Params (M)	FLOPs (G)	FPS
YOLOv3	73.1	37.0	61.5	154.6	40
YOLOv3-tiny	54	22.4	8.6	12.9	160
YOLOv5-s	69.8	36.0	7.0	15.8	96
MobileNetv3-YOLOv5	71.9	36.6	5.0	11.3	72
ShuffleNetv2-YOLLOv5	63.7	31.5	3.8	8.0	83
GhostNet-YOLOv5	73.2	36.6	4.7	7.6	74
YOLOv7-tiny	69.3	32.6	6.0	13.1	99
YOLOv8s	71.8	37.2	6.2	12.1	106
SDD-YOLO	76.1	40.3	3.4	6.4	121

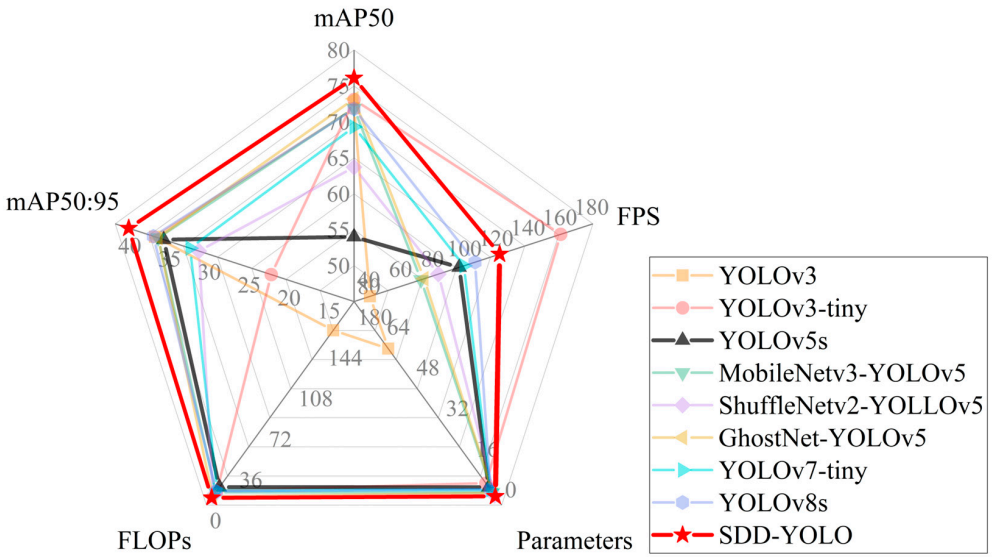

Figure 10. The performance comparison of different detection algorithms.

The SDD-YOLO method proposed in this study achieved a $mAP_{50:95}$ of 40.3%, surpassing all other classical methods, while significantly reducing complexity compared to all other network models, with Params and FLOPs being only 3.4M and 6.4G, respectively. Although YOLOv3-tiny achieved the highest FPS, its detection performance was poor, with a $mAP_{50:95}$ of only 22.4%. The SDD-YOLO proposed in this study achieved the best results in terms of detection accuracy, parameter count, and computational complexity, outperforming all lightweight networks and most mainstream networks. Compared to the baseline YOLOv5s, the proposed SDD-YOLO reduced parameter count and computational complexity by 51.4% and 59.5%, respectively, while increasing speed by 2.1 times. Moreover, the detection performance of ShuffleNetv2-YOLOv5, MobileNetv3-YOLOv5, and GhostNet-YOLOv5, which replaced the backbone network, was lower than that of SDD-YOLO proposed in this study.

Figure 10 emphasizes the comprehensive performance of the proposed SDD-YOLO model, achieving the highest mean average precision (mAP) while maintaining low parameter count and lightweightness, highlighting its superior performance in strip steel surface defect detection compared to the other eight models.

To evaluate the performance of the proposed SDD-YOLO method in detecting different defect types, this study compared it with multiple other models. Table 6 and Figure 11 present the proposed SDD-YOLO method's performance compared to other classical methods in terms of average precision (AP).

Table 6. Comparative analysis of AP performance between SDD-YOLO and classical object detection methods.

Method	Cr (%)	In (%)	Pa (%)	Ps (%)	Rs (%)	Sc (%)
YOLOv3	50.1	85.1	90.3	90.4	54.6	88.2
YOLOv3-tiny	54.0	84.2	92.2	89.5	55.1	87.5
YOLOv5-s	51.6	85.3	92.7	88.9	55.4	88.6
MobileNetv3-YOLOv5	52.7	83.1	91.2	90.5	52.6	89.5
ShuffleNetv2-YOLLOv5	54.1	84.2	91.8	90.3	51.5	89.1
GhostNet-YOLOv5	53.2	83.3	92.5	91.1	54.5	90.3
YOLOv7-tiny	53.6	84.1	92.4	90.7	53.6	86.9
YOLOv8s	52.4	86.4	91.9	91.7	52.4	90.2
SDD-YOLO	58.1	88.1	94.9	93.4	61.7	91.3

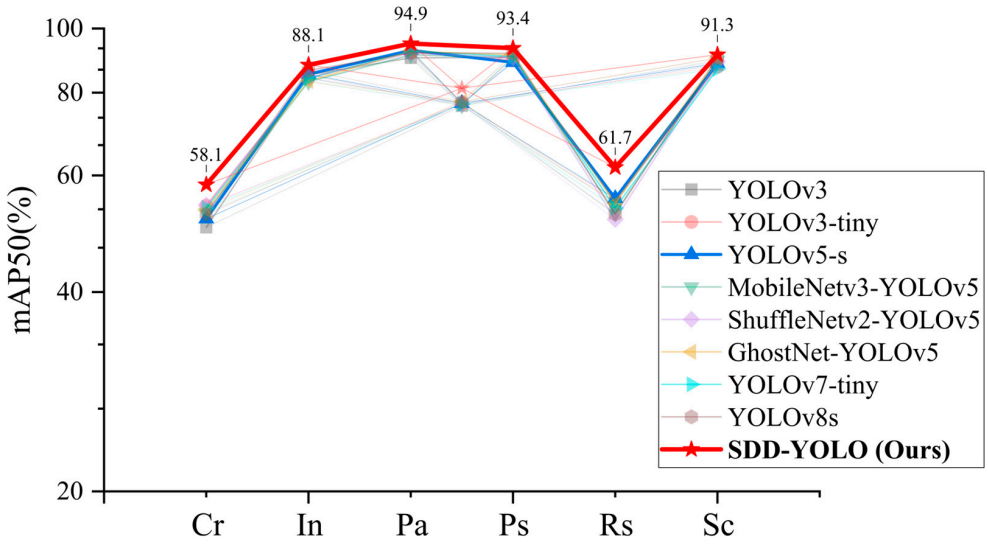

Figure 11. Comparative analysis of AP performance between SDD-YOLO and classical object detection methods.

Table 6 shows that the SDD-YOLO model's defect detection capabilities for each category surpass all classical algorithms, demonstrating that the proposed CGH module and MCFF effectively enhance the model's ability in strip steel surface defect detection tasks.

Figure 11 illustrates the performance comparison between SDD-YOLO and classical object detection methods. By visualizing the centroids of the average precision (AP) values of the eight models, it is evident that the SDD-YOLO method significantly outperforms other methods in overall detection performance. Moreover, the AP values for each category highlight the proposed method's capability in detecting specific surface defect categories, with SDD-YOLO outperforming all other classical methods, especially in the Cr and Rs categories, where the AP values increased by 8% and 7.1%, respectively.

4.4. Robustness Testing

In the acquisition of metal surface images under real-world conditions, various environmental factors such as overexposure, dim lighting, and image blurriness often exert influence. To assess the adaptability of the SDD-YOLO model to these scenarios, this study subjected the dataset's images to three types of image interference processing: high

exposure, low brightness, and the addition of Gaussian noise. Partial dataset images before and after image interference processing are depicted in Figure 12.

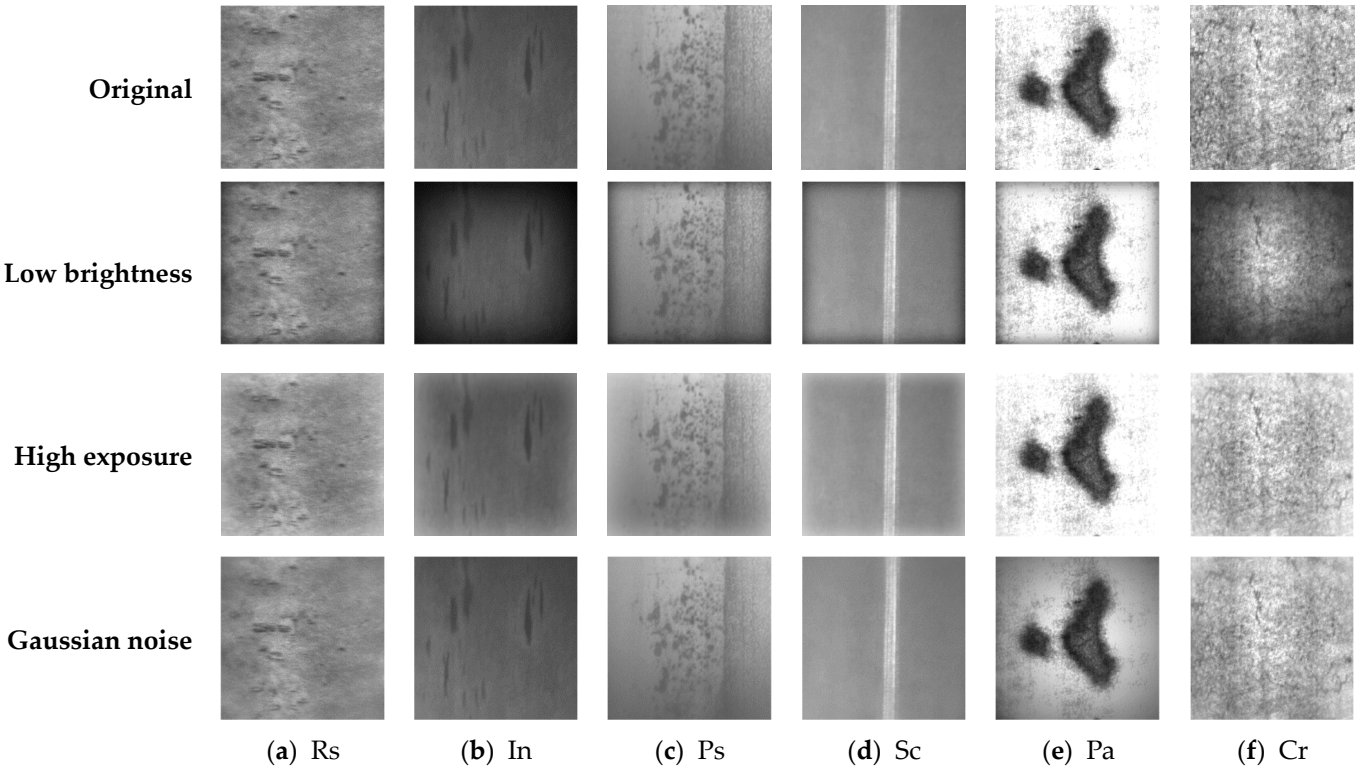

Figure 12. Partial dataset images before and after image interference processing.

For comparative analysis of the robustness and generalization capabilities of the method proposed in this study and the benchmark YOLOv5s, both methods were separately applied to the dataset after image interference processing, yielding the results shown in Table 7.

Table 7. Comparative analysis of strip steel surface defect detection performance between SDD-YOLO and YOLOv5s on dataset before and after image interference processing.

Data	Proposed SDD-YOLO			YOLOv5s		
	mAP_{50}	$mAP_{50:95}$	FPS	mAP_{50}	$mAP_{50:95}$	FPS
Original	76.1	40.3	121	69.8	36.0	96
Low brightness	69.2	33.1	128	57.3	26.2	107
High exposure	72.5	36.9	126	62.3	29.1	104
Gaussian noise	71.7	35.7	131	62.2	28.5	112
All processed data	71.1	35.2	129	59.1	27.5	108

In the processed dataset, both YOLOv5s and SDD-YOLO exhibit a decline in performance. Specifically, YOLOv5s experiences reductions of 15.35% and 23.61% in mAP_{50} and $mAP_{50:95}$, respectively, while SDD-YOLO experiences reductions of 6.57% and 12.66% in mAP_{50} and $mAP_{50:95}$, respectively. Despite the susceptibility of the SDD-YOLO proposed in this study to missed detections and false detections when handling blurred images, overall, the model demonstrates its concurrent detection capability, achieving a final mAP of 71.1%. Compared to using the benchmark YOLOv5s model, SDD-YOLO achieves a 12% improvement in detecting interfered images, even surpassing the accuracy of YOLOv5s in detecting undisturbed images. Taking all factors into consideration, the improved SDD-YOLO model presented in this study exhibits stronger robustness and generalization capabilities than the benchmark YOLOv5s model.

5. Conclusions

In this study, a lightweight strip steel surface defect detection network named SDD-YOLO is introduced, based on YOLOv5s, to address the challenge of low precision in strip steel surface defect detection while maintaining network lightweightness. To enhance computational efficiency and detection accuracy, this study designs CGH and MCFF modules and adopts a BiFPN structure to fuse features of different scales, thereby enhancing the detector's adaptability to targets of various scales. Additionally, the CARAFE module is utilized to replace bilinear interpolation upsampling, thereby enlarging the receptive field of upsampling to extract more image features and improve model performance. On the NEU-DET dataset, SDD-YOLO achieves an mAP of 76.1% with a model size of only 3.4MB and FLOPs of 6.4G, marking an improvement of 6.3 percentage points compared to YOLOv5s, with an FPS of 121. Compared with YOLOv5s, the model size and computational load are reduced by 51.4% and 59.5%, respectively. Through ablation experiments and comprehensive performance comparisons with state-of-the-art methods, the superior performance and generalization ability of the SDD-YOLO method in strip steel surface defect detection tasks are validated. Additionally, potential issues in practical application are thoroughly considered in this study by introducing image interference techniques such as high exposure, low brightness, and adding Gaussian noise to the dataset, thereby conducting a robustness analysis of the SDD-YOLO method. The analysis results comprehensively demonstrate the method's real-time capability and strong generalization ability. In conclusion, the proposed SDD-YOLO exhibits characteristics of high precision, efficiency, and lightweightness, making it suitable for deployment in real-world production environments for real-time strip steel surface defect localization and detection, thereby enhancing strip steel production efficiency and quality. In our future endeavors, we will focus on further optimizing the algorithm to achieve higher accuracy, faster detection speed, and lower model complexity.

Author Contributions: Conceptualization, Y.W. and Z.L.; Data curation, Y.W.; Formal analysis, Y.W. and R.C.; Funding acquisition, M.D.; Investigation, Y.W. and R.C.; Methodology, Y.W. and R.C.; Project administration, M.D.; Resources, Y.W. and M.Y.; Software, Y.W.; Supervision, Z.L. and M.D.; Validation, Y.W. and R.C.; Visualization, R.C. and M.Y.; Writing—original draft, Y.W. and R.C.; Writing—review and editing, Y.W., R.C., Z.L. and M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a special grant from the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2023A1515011326, the program for scientific research start-up funds of Guangdong Ocean University under Grant No. 060302102101, Guangdong Provincial Science and Technology Innovation Strategy under Grant No. pdjh2023b0247, National College Students Innovation and Entrepreneurship Training Program under Grant No. 202310566022, and Guangdong Ocean University Undergraduate Innovation Team Project under Grant No. CXTD2023014.

Data Availability Statement: The dataset that supports the findings of this study are available in [NEU-DET] at "http://faculty.neu.edu.cn/songkc/en/zhym/263264/list/index.htm (accessed on 29 May 2024)".

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

References

1. Kim, S.; Kim, W.; Noh, Y.-K.; Park, F.C. Transfer learning for automated optical inspection. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 2517–2524.
2. Lv, X.; Duan, F.; Jiang, J.J.; Fu, X.; Gan, L.J.S. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* **2020**, *20*, 1562. [CrossRef] [PubMed]
3. Czimmermann, T.; Ciuti, G.; Milazzo, M.; Chiurazzi, M.; Roccella, S.; Oddo, C.M.; Dario, P. Visual-Based Defect Detection andClassification Approaches for Industrial Applications—A SURVEY. *Sensors* **2020**, *20*, 1459. [CrossRef] [PubMed]
4. Fang, X.; Luo, Q.; Zhou, B.; Li, C.; Tian, L. Research Progress of Automated Visual Surface Defect Detection for Industrial Metal Planar Materials. *Sensors* **2020**, *20*, 5136. [CrossRef] [PubMed]

5. Xie, L.; Baskaran, P.; Ribeiro, A.L.; Alegria, F.C.; Ramos, H.G. Classification of Corrosion Severity in SPCC Steels Using Eddy Current Testing and Supervised Machine Learning Models. *Sensors* **2024**, *24*, 2259. [CrossRef] [PubMed]
6. Zou, Y.; Fan, Y. An Infrared Image Defect Detection Method for Steel Based on Regularized YOLO. *Sensors* **2024**, *24*, 1674. [CrossRef] [PubMed]
7. Yousaf, J.; Harseno, R.W.; Kee, S.-H.; Yee, J.-J. Evaluation of the Size of a Defect in Reinforcing Steel Using Magnetic Flux Leakage (MFL) Measurements. *Sensors* **2023**, *23*, 5374. [CrossRef] [PubMed]
8. Subramanyam, V.; Kumar, J.; Singh, S.N. Temporal synchronization framework of machine-vision cameras for high-speed steel surface inspection systems. *J. Real-Time Image Process.* **2022**, *19*, 445–461. [CrossRef]
9. Kang, Z.; Yuan, C.; Yang, Q. The fabric defect detection technology based on wavelet transform and neural network convergence. In Proceedings of the 2013 IEEE International Conference on Information and Automation (ICIA), Yinchuan, China, 26–28 August 2013; pp. 597–601.
10. Hu, S.; Li, J.; Fan, H.; Lan, S.; Pan, Z. Scale and pattern adaptive local binary pattern for texture classification. *Expert Syst. Appl.* **2024**, *240*, 122403. [CrossRef]
11. Abouzahir, S.; Sadik, M.; Sabir, E. Bag-of-visual-words-augmented Histogram of Oriented Gradients for efficient weed detection. *Biosyst. Eng.* **2021**, *202*, 179–194. [CrossRef]
12. Shayeste, H.; Asl, B.M. Automatic seizure detection based on Gray Level Co-occurrence Matrix of STFT imaged-EEG. *Biomed. Signal Process. Control* **2023**, *79*, 104109. [CrossRef]
13. Anter, A.M.; Abd Elaziz, M.; Zhang, Z. Real-time epileptic seizure recognition using Bayesian genetic whale optimizer andadaptive machine learning. *Future Gener. Comput. Syst.* **2022**, *127*, 426–434. [CrossRef]
14. Malek, A.S.; Drean, J.; Bigue, L.; Osselin, J. Optimization of automated online fabric inspection by fast Fourier transform (FFT) and cross-correlation. *Text. Res. J.* **2013**, *83*, 256–268. [CrossRef]
15. Zhou, X.; Wang, Y.; Zhu, Q.; Mao, J.; Xiao, C.; Lu, X.; Zhang, H. A Surface Defect Detection Framework for Glass Bottle Bottom Using Visual Attention Model and Wavelet Transform. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2189–2201. [CrossRef]
16. Ma, J.; Wang, Y.; Shi, C.; Lu, C. Fast Surface Defect Detection Using Improved Gabor Filters. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1508–1512.
17. Kulkarni, R.; Banoth, E.; Pal, P. Automated surface feature detection using fringe projection: An autoregressive modeling-based approach. *Opt. Laser Eng.* **2019**, *121*, 506–511. [CrossRef]
18. Hao, M.; Zhou, M.; Jin, J.; Shi, W. An Advanced Superpixel-Based Markov Random Field Model for Unsupervised Change Detection. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1401–1405. [CrossRef]
19. Liu, J.; Cui, G.; Xiao, C. A Real-time and Efficient Surface Defect Detection Method Based on YOLOv4. *J. Real-Time Image Process.* **2023**, *20*, 1–15. [CrossRef]
20. Chen, L.; Wu, X.; Sun, C.; Zou, T.; Meng, K.; Lou, P. An intelligent vision recognition method based on deep learning for pointer meters. *Meas. Sci. Technol.* **2023**, *34*, 055410. [CrossRef]
21. Huang, Z.; Hu, H.; Shen, Z.; Zhang, Y.; Zhang, X. Lightweight edge-attention network for surface-defect detection of rubber seal rings. *Meas. Sci. Technol.* **2022**, *33*, 085401. [CrossRef]
22. Lin, Z.; Ye, H.; Zhan, B.; Huang, X. An Efficient Network for Surface Defect Detection. *Appl. Sci.* **2020**, *10*, 6085. [CrossRef]
23. Li, H.; Wang, F.; Liu, J.; Song, H.; Hou, Z.; Dai, P. Ensemble model for rail surface defects detection. *PLoS ONE* **2023**, *18*, e0292773. [CrossRef] [PubMed]
24. Zhou, C.; Lu, Z.; Lv, Z.; Meng, M.; Tan, Y.; Xia, K.; Liu, K.; Zuo, H. Metal surface defect detection based on improved YOLOv5. *Sci. Rep.* **2023**, *13*, 20803. [CrossRef] [PubMed]
25. Zhang, Y.; Shen, S.; Xu, S. Strip steel surface defect detection based on lightweight YOLOv5. *Front. Neurorobot.* **2023**, *17*, 1263739. [CrossRef] [PubMed]
26. Lv, B.; Duan, B.; Zhang, Y.; Li, S.; Wei, F.; Gong, S.; Ma, Q.; Cai, M. Research on Surface Defect Detection of Strip Steel Based on Improved YOLOv7. *Sensors* **2024**, *24*, 2667. [CrossRef] [PubMed]
27. Li, Y.; Xu, S.; Zhu, Z.; Wang, P.; Li, K.; He, Q.; Zheng, Q. EFC-YOLO: An Efficient Surface-Defect-Detection Algorithm for Steel Strips. *Sensors* **2023**, *23*, 7619. [CrossRef] [PubMed]
28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
29. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2778–2788.
30. Li, C.; Yan, H.; Qian, X.; Zhu, S.; Zhu, P.; Liao, C.; Tian, H.; Li, X.; Wang, X.; Li, X. A domain adaptation YOLOv5 model for industrial defect inspection. *Measurement* **2023**, *213*, 112725. [CrossRef]
31. Shi, H.; Zhao, D. License Plate Recognition System Based on Improved YOLOv5 and GRU. *IEEE Access* **2023**, *11*, 10429–10439. [CrossRef]
32. Lawal, O.M. YOLOv5-LiNet: A lightweight network for fruits instance segmentation. *PLoS ONE* **2023**, *18*, e0282297. [CrossRef] [PubMed]
33. Cardellicchio, A.; Solimani, F.; Dimauro, G.; Petrozza, A.; Summerer, S.; Cellini, F.; Renò, V. Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors. *Comput. Electron. Agric.* **2023**, *207*, 107757. [CrossRef]

34. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Machine Learning and Knowledge Discovery in Databases, Proceedings of the European Conference, ECML PKDD 2022, Grenoble, France, 19–23 September 2022*; Part III; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 443–459.
35. Sergey, I.; Christian, S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning—Volume 37, JMLR.org, Lille, France, 7–9 July 2015; pp. 448–456.
36. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]
37. Kim, D.; Park, S.; Kang, D.; Paik, J. Improved center and scale prediction-based pedestrian detection using convolutional block. In Proceedings of the 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 8–11 September 2019; pp. 418–419.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
39. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
40. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
41. Wang, J.; Pan, Q.; Lu, D.; Zhang, Y. An Efficient Ship-Detection Algorithm Based on the Improved YOLOv5. *Electronics* **2023**, *12*, 3600. [CrossRef]
42. Liu, H.; Duan, X.; Chen, H.; Lou, H.; Deng, L. DBF-YOLO: UAV Small Targets Detection Based on Shallow Feature Fusion. *IEEJ Trans. Electr. Electron. Eng.* **2023**, *18*, 605–612. [CrossRef]
43. Zhang, X.; Feng, Y.; Zhang, S.; Wang, N.; Mei, S. Finding Nonrigid Tiny Person With Densely Cropped and Local Attention Object Detector Networks in Low-Altitude Aerial Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 4371–4385. [CrossRef]
44. Liu, P.; Wang, Q.; Zhang, H.; Mi, J.; Liu, Y. A Lightweight Object Detection Algorithm for Remote Sensing Images Based on Attention Mechanism and YOLOv5s. *Remote Sens.* **2023**, *15*, 2429. [CrossRef]
45. Zha, W.; Hu, L.; Sun, Y.; Li, Y. ENGD-BiFPN: A remote sensing object detection model based on grouped deformable convolution for power transmission towers. *Multimed. Tools Appl.* **2023**, *82*, 45585–45604. [CrossRef]
46. Lu, X.; Lu, X. An efficient network for multi-scale and overlapped wildlife detection. *Signal Image Video Process.* **2023**, *17*, 343–351. [CrossRef]
47. Jiang, M.; Song, L.; Wang, Y.; Li, Z.; Song, H. Fusion of the YOLOv4 network model and visual attention mechanism to detect low-quality young apples in a complex environment. *Precis. Agric.* **2022**, *23*, 559–577. [CrossRef]
48. Ma, S.; Zhang, X.; Jia, C.; Zhao, Z.; Wang, S.; Wang, S. Image and video compression with neural networks: A review. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1683–1698. [CrossRef]
49. Lan, R.; Sun, L.; Liu, Z.; Lu, H.; Pang, C.; Luo, X. MADNet: A fast and lightweight network for single-image super resolution. *IEEE Trans. Cybern.* **2020**, *51*, 1443–1453. [CrossRef] [PubMed]
50. Liu, C.; Gao, H.; Chen, A. A real-time semantic segmentation algorithm based on improved lightweight network. In Proceedings of the 2020 International Symposium on Autonomous Systems (ISAS), Guangzhou, China, 6–8 December 2020; pp. 249–253.
51. Bao, Y.; Song, K.; Liu, J.; Wang, Y.; Yan, Y.; Yu, H.; Li, X. Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5011111. [CrossRef]
52. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
53. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. Carafe: Content-aware reassembly of features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
54. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.

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.