applied sciences

MDPI

Article

FSNB-YOLOV8: Improvement of Object Detection Model for Surface Defects Inspection in Online Industrial Systems

Jun Li [1,*], Jinglei Wu [2] and Yanhua Shao [3]

[1] School of Computer Science, Beijing Information Science and Technology University, Beijing 100192, China
[2] National Institute of Biological Sciences, Beijing 102206, China; bistuwjl2023@bistu.edu.cn
[3] National Computer System Engineering Research Institute of China, Beijing 100083, China; stephen_yanhuashao@outlook.com
* Correspondence: lijun@bistu.edu.cn

Abstract: The current object detection algorithm based on CNN makes it difficult to effectively capture the characteristics of subtle defects in online industrial product packaging bags. These defects are often visually similar to the texture or background of normal product packaging bags, and the model cannot effectively distinguish them. In order to deal with these challenges, this paper optimizes and improves the network structure based on YOLOv8 to achieve accurate identification of defects. First, in order to solve the long-tail distribution problem of data, a fuzzy search data enhancement algorithm is introduced to effectively increase the number of samples. Secondly, a joint network of FasterNet and SPD-Conv is proposed to replace the original backbone network of YOLOv8, which effectively reduces the computing load and improves the accuracy of defect identification. In addition, in order to further improve the performance of multiscale feature fusion, a weighted bidirectional feature pyramid network (BiFPN) is introduced, which effectively enhances the model's ability to detect defects at different scales through the fusion of deep information and shallow information. Finally, in order to reduce the sensitivity of the defect position deviation, the NWD loss function is used to optimize the positioning performance of the model better and reduce detection errors caused by position errors. Experimental results show that the FSNB_YOLOv8 model proposed in this paper can reach 98.8% mAP50 accuracy. This success not only verifies the effectiveness of the optimization and improvement of this article's model but also provides an efficient and accurate solution for surface defect detection of industrial product packaging bags on artificial assembly systems.

Keywords: industrial products systems; image processing; deep learning; small object detection; artificial information visualization

Citation: Li, J.; Wu, J.; Shao, Y. FSNB-YOLOV8: Improvement of Object Detection Model for Surface Defects Inspection in Online Industrial Systems. *Appl. Sci.* **2024**, *14*, 7913. https://doi.org/10.3390/app14177913

Academic Editor: Thomas Lindner

Received: 10 July 2024
Revised: 5 August 2024
Accepted: 3 September 2024
Published: 5 September 2024

1. Introduction

The volume of packaging circulation data in industrial product systems is substantial and exhibits a high rate of repetition, with only a small amount of defective data being used for circulation tracing and quality management [1]. Detecting genuine defective data from massive real-time packaging systems and placing them in a data monitoring and governance pool is a challenging problem. The accuracy and timeliness [2] of data detection directly affect the improvement level of industrial product packaging line monitoring and governance, and even the entire industrial production and logistics quality control system [3].

In real-time production lines of industrial packaging bags, accurate and efficient detection of subtle defects [4] in the packaging material holds crucial significance for ensuring the quality and safety of the packaged goods [5]. However, due to the scarcity of defect samples in practical production systems, the data samples exhibit a long-tailed distribution [6], significantly limiting the generalization capabilities of models and subsequently compromising the effectiveness of actual detections [7]. Therefore, effective data augmentation is essential to enhance object detection performance [8]. Some studies have attempted

to combine traditional algorithms with object detection [9] by utilizing techniques such as affine transformations [10], flipping [11], cropping [12], and padding [13] to enhance defect samples related to industrial product packaging bags. Additionally, research has explored the integration of deep learning with object detection [14], enriching defect samples through algorithms like Mosaic [15], Mixup [16], and Cutout [17], specifically tailored for the challenges encountered in industrial online packaging bag production.

Nevertheless, these methods often have limitations and rely heavily on parameter adjustments, resulting in suboptimal enhancement effects in industrial product packaging bags. Furthermore, mainstream object detection algorithms, such as Faster R-CNN [18], CenterNet [19], and the YOLO [20] series, have demonstrated remarkable performance in many fields. However, they face significant challenges when dealing with the detection of subtle defects in industrial product packaging bags. During downsampling [21], these algorithms often lose crucial feature information due to the similarity between defects and the background texture [22] of the packaging, making it difficult to effectively capture and identify defects. Additionally, current network structures lack sufficient capabilities for multiscale feature [23] fusion, resulting in limited recognition abilities for subtle defects. Moreover, existing algorithms are highly sensitive to positional deviations of small targets [24], and minor changes in the target location can significantly impact the detection results, further complicating the detection of small defects [25] in industrial product packaging systems.

Therefore, to address the challenges of detecting minute defects on the surface of industrial product packaging bags in industrial assembly systems, we propose the small-sample object detection model FSNB_YOLOv8, which is based on an improved YOLOv8 [26]. This model alleviates the issue of long-tailed data distribution and enhances the generalization capability of the model by introducing a fuzzy search interactive data augmentation algorithm [27]. At the same time, the network structure is optimized by replacing the original backbone network with a combination of FasterNet [28] and SPD-Conv [29], reducing the computational complexity and improving the detection accuracy for small target defects. Furthermore, a weighted bidirectional feature pyramid network [30] is introduced to enhance the performance of multiscale feature fusion and improve detection accuracy. Additionally, the NWD [31] loss function is employed to optimize positioning performance and reduce detection errors caused by positional deviations, particularly for small and subtle defects in industrial product packaging systems.

The primary contributions of this paper are as follows:

- To address the challenge of the long-tailed distribution of industrial product packaging bag data in practical industrial scenarios, we introduce an interactive data augmentation algorithm based on fuzzy search. This algorithm effectively increases the number of defective samples by intelligently generating and enhancing them, thereby alleviating the issue of data imbalance and enhancing the generalization capabilities of the model. By leveraging this approach, we can improve the performance of defect detection systems in real-world industrial settings, where data distribution is often imbalanced.
- To address the high computational complexity of the C2F module, we have designed a joint network incorporating FasterNet and SPD-Conv, which replaces the original backbone network of YOLOv8. This improvement aims to reduce the computational load and network redundancy while enhancing the accuracy of small target defect detection. By optimizing the network structure, FSNB_YOLOv8 can better adapt to the computational resource constraints of edge devices while maintaining its performance.
- To further enhance the performance of multiscale feature fusion, we introduce the weighted bi-directional feature pyramid network (BiFPN). This network effectively enhances the model's ability to detect defects at different scales by integrating deep and shallow information, thereby improving the accuracy and stability of the detection process.

- To reduce the sensitivity to defects' positional deviations, this paper employs the Normalized Wasserstein Distance (NWD) loss function. This function can more accurately measure the distance between the predicted bounding boxes and actual boxes, optimizing the model's localization performance and reducing detection errors caused by positional deviations.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work on small-sample defect detection and image enhancement. Section 3 details the specific improvement measures proposed in this article. Section 4 presents the experimental results and analysis. Finally, Section 5 discusses and summarizes the results of this study.

2. Related Work

Research on defect detection based on deep learning often relies on large-scale datasets to train network models, enabling the precise extraction of defect features [32]. However, in industrial pipeline practices, obtaining defect data often faces numerous challenges, such as high human and time costs, which render traditional large-scale data collection methods impractical. To address this issue, researchers have actively explored various strategies, including metric learning, data augmentation, transfer learning, and model fine-tuning, to efficiently perform surface defect detection using deep learning with limited sample data [33]. By leveraging these techniques, it is feasible to achieve accurate defect detection in industrial settings, despite the scarcity of labeled data. This approach holds promise for enhancing quality control and improving production efficiency in various manufacturing industries.

Ling et al. [34] designed a deep Siamese semantic segmentation network that combines the similarity measurement of the Siamese network with the encoder-decoder semantic segmentation network for PCB welding defect detection. This approach further alleviates the overfitting problem caused by insufficient defect samples. Li [35] proposed a weakly supervised machine vision detection method based on artificial defect simulation that applied artificial datasets to deep learning recognition algorithms and fine-tuned the initial model for retraining. Liu [36] introduced a knowledge reuse strategy to train a convolutional neural network (CNN) model. This strategy incorporates model-based transfer learning and data augmentation to achieve high accuracy with limited training samples. Li [37] utilized a single-stage detection model, YOLO, to detect steel surface defects. By integrating shallow features, they improved the detection performance of YOLO for small-sample defects. Liu [38] proposed a fabric defect detection framework based on Generative Adversarial Networks (GANs). This framework trains multilevel GANs to synthesize realistic defects in new defect-free samples and incorporates the generated defects into specific locations to better detect defects under different conditions. Zhang [39] developed a semi-supervised Generative Adversarial Network (SSGAN) that can achieve more accurate segmentation results at the pixel level. It leverages unlabeled images to enhance segmentation performance and reduce the burden of data-labeling tasks. He [40] introduced a novel method based on weakly supervised deep learning that designs an encoder for feature extraction and two decoders for two related tasks. This approach accurately segments and locates defects on textured surfaces.

In this paper, we address the challenges of detecting subtle defects on the surface of industrial product packaging bags in industrial assembly systems. To this end, we introduce a fuzzy search interactive data augmentation algorithm that effectively alleviates the problem of long-tailed data distribution and enhances the generalization capabilities of our model. Furthermore, we optimize the network structure by replacing the original backbone network with a combined FasterNet and SPD-Conv architecture. This optimization not only reduces the computational complexity but also improves the accuracy of detecting small target defects. Additionally, we introduce a weighted bidirectional feature pyramid network (BiFPN) to enhance the multiscale feature fusion performance, further boosting detection accuracy. Lastly, we adopt the Normalized Wasserstein Distance (NWD)

loss function to optimize localization performance, minimizing detection errors caused by positional deviations. By implementing these comprehensive strategies, our model effectively addresses the challenges of defect detection in industrial assembly systems, providing a robust and efficient solution for identifying subtle defects on the surface of industrial product packaging bags.

3. Proposed Method

To address the challenges associated with the detection of minute defects on the surface of industrial product packaging bags in industrial assembly systems, we propose FSNB_YOLOv8, a small-sample object detection model based on an enhanced version of YOLOv8.

This model introduces a fuzzy search interactive data augmentation algorithm that effectively mitigates the issue of long-tailed data distribution and enhances the model's generalization capabilities. Furthermore, the network structure is optimized by replacing the original backbone network with a combined FasterNet and SPD-Conv architecture. This optimization not only reduces computational complexity but also improves the accuracy of detecting small target defects. Additionally, we introduce a weighted bidirectional feature pyramid network (BiFPN) to enhance the multiscale feature fusion performance, further enhancing the detection accuracy. Lastly, we employ the Normalized Wasserstein Distance (NWD) loss function to optimize localization performance, minimizing detection errors caused by positional deviations. Through these comprehensive strategies, the FSNB_YOLOv8 model effectively addresses the challenges of defect detection in industrial assembly systems, providing a robust and efficient solution for identifying minute defects on the surface of industrial product packaging bags.

3.1. Interactive Fuzzy Search Data Augmentation Algorithm

The lack of a universal dataset for industrial product packaging bag defect samples in the industrial sector poses a significant challenge. Creating a custom dataset is a crucial step in applying object detection models for defect detection in industrial product packaging bags. However, in practical industrial scenarios, the overwhelming majority of samples on the assembly line are normal, while defect samples are extremely rare. This results in a long-tailed distribution of data, which is highly unfavorable for model training and optimization. To address these issues, this paper introduces an interactive defect fuzzy search algorithm that generates a large quantity of high-quality simulated defect data. This approach effectively addresses the problem of insufficient defect detection capabilities due to the scarcity of defect samples.

The experimental dataset consists of real-world sample data collected from actual industrial assembly systems. Based on this, a significant amount of the simulated defect dataset is generated offline. According to Equations (1)–(3), the following steps are taken to obtain a batch of simulated defect data: Initially, we identify a point center proximate to the defect's centroid on the labeled sample and establish a 3×3 pixel search neighborhood $U(i, j)$ centered on this point.

Modeled as follows:

$$U(i,j) = \{(x,y) | x_c - 1 \leq x \leq x_c + 1, y_c - 1 \leq y \leq y_c + 1\} \quad (1)$$

$U(i, j)$ is the pixel search neighborhood, (x, y) is the point where the marked sample is close to the center of the defect.

Subsequently, we assess the similarity S of each point $X(i, j)$ within $U(i, j)$ by analyzing the grayscale values and documenting the corresponding positional coordinates. This enables the identification of image regions exhibiting black spots and aluminum foil damage on the industrial products' packaging bags, as outlined in Equation (2).

Modeled as follows:

$$S(i,j) = \frac{1}{N} \sum_{k,l \in U(i,j)} \left| G(k,l) - G_{avg} \right| \quad (2)$$

$S(i,j)$ is the similarity of each point calculated from the gray value of each pixel, and $G(k,l)$ is the gray value of a single pixel in the neighborhood. G_{avg} is the average gray value of its neighborhood, and N is the number of pixels in the neighborhood $U(i,j)$. This process can eliminate the influence of neighborhood size and make similarity measures comparable.

Finally, we extract the pertinent defective subimage regions from the original image and manipulate them through operations such as flipping and scaling. These manipulated defective sub-images are then seamlessly integrated into random locations on normal samples, generating a substantial quantity of defective samples for industrial product packaging bags, as specified in Equation (3).

Modeled as follows:

$$F(x,y) = \alpha \cdot D(x,y) + (1-\alpha) \cdot N(x,y) \quad (3)$$

$F(x,y)$ is the pixel value at the corresponding coordinates of the final generated industrial product packaging bag sample containing defects, $D(x,y)$ is the pixel value of the original defect sub-image at the corresponding coordinates, $N(x,y)$ is the pixel value of the normal sample at the corresponding coordinates, and α is a mixing coefficient between 0 and 1, used to control the relative contributions of the original defect sub-image $D(x, y)$ and the normal sample $N(x, y)$ when generating a new sample $F(x, y)$, as illustrated in Figure 1. If industrial product packaging encounters different color varieties during production, the grayscale image collection and detection method shown in Figure 1 can be generalized to adapt to various types of varieties.

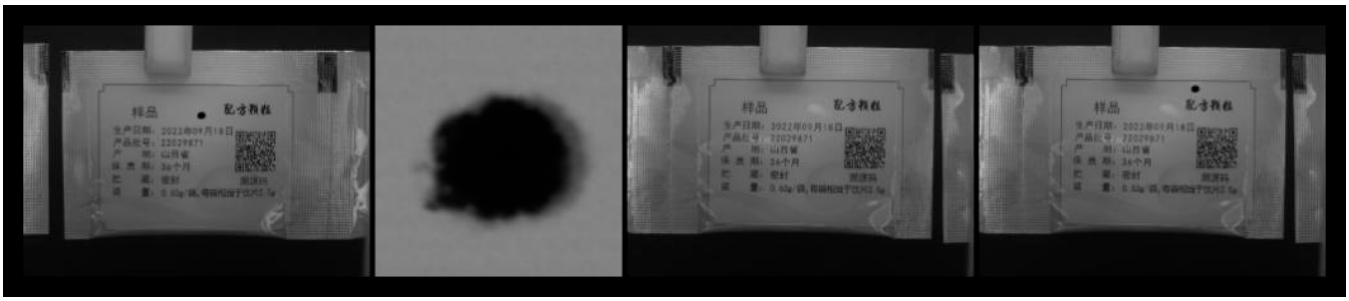

Figure 1. An example of defect simulation data generation for industrial product packaging bags. By locating defects and generating defect targets for the first sub-image, the second sub-image is formed, and the fourth new defect complete sample is formed based on the second sub-image and the third normal image. This generation method can generate a large amount of high-quality and diverse defect simulation data. The Chinese characters on the surface of the medicine bag indicate the name of the medicine bag particles and the specific packaging requirements.

3.2. FasterNet Lightweight Network

After being processed by deep neural networks, images often produce a large number of redundant feature maps. The backbone network of YOLOv8 performs convolution on all feature maps channel-by-channel, leading to excessive redundant calculations, which affect the detection efficiency of the model. To address this issue, this paper proposes replacing the Bottleneck module in the C2f section of the YOLOv8 backbone network with the FasterBlock module from the FasterNet network architecture. This replacement aims to reduce redundant calculations and enhance the detection efficiency of the model, as illustrated in Figure 2.

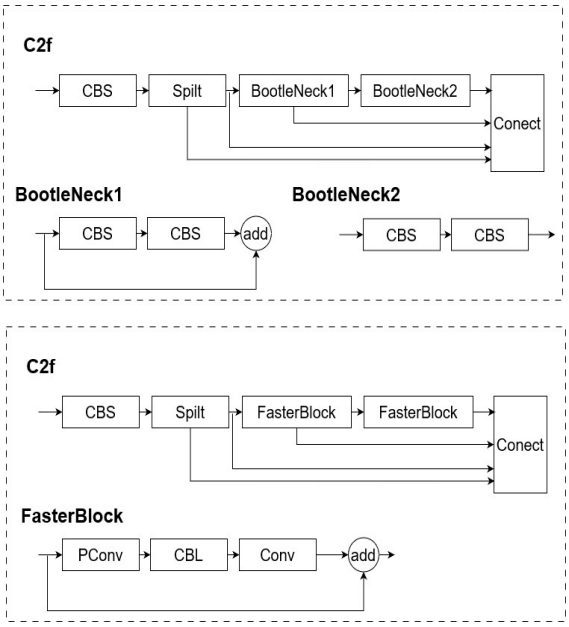

Figure 2. Comparison within the YOLOv8 object detection framework, showcasing the original C2f module versus the enhanced C2f-Faster module, where the latter incorporates FasterBlock in place of the former. The introduction of the FasterBlock module further accelerates the defect detection process.

3.2.1. The Architecture of FasterNet Network

FasterNet is an efficient neural network architecture designed to enhance computational speed without compromising accuracy. It demonstrates excellent performance in handling visual tasks. It employs a novel technique called partial convolution (PConv) to reduce redundant computations and memory accesses while accelerating neural networks to minimize the time required for model inference. PConv achieves this by processing only a subset of the input channels, thereby reducing the computational burden and memory footprint. Leveraging this advantage, FasterNet achieves faster runtimes on various devices compared to other networks while maintaining high accuracy.

The overall architecture of FasterNet comprises four hierarchical stages, with each stage consisting of a series of FasterNet blocks preceded by embedding or merging layers. The last three layers are used for feature classification. Within each FasterNet block, a PConv layer is followed by two pointwise convolution (PWConv) layers. To preserve feature diversity and achieve lower latency, normalization and activation layers are placed only after the intermediate layers. Figure 3 illustrates the overall architecture of FasterNet.

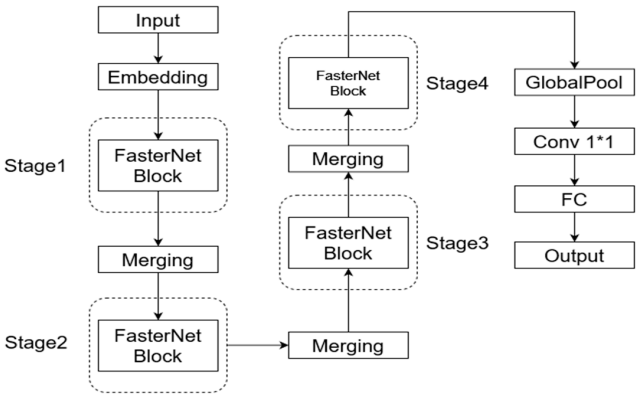

Figure 3. Illustration of the FasterNet neural network architecture, a lightweight design that minimizes redundant computations and memory access to extract spatial features more efficiently.

3.2.2. Partial Convolution (PConv)

Partial Convolution (PConv) significantly enhances the computational efficiency by incorporating a mask into the convolution operation, as shown in Equation (4). Instead of applying convolution comprehensively, as in traditional methods, PConv performs the operation only on a subset of the input feature map. This approach allows PConv to reduce unnecessary computations and memory accesses, effectively ignoring redundant portions of the input. A comparison between partial convolution (PConv) and ordinary convolution is illustrated in Figure 4.

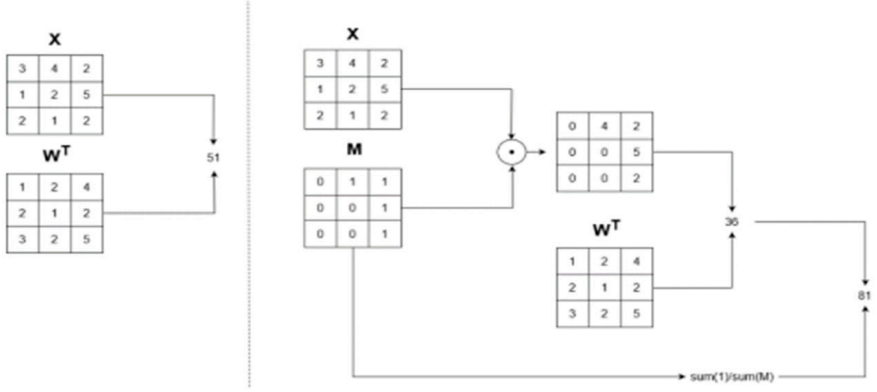

Figure 4. Example comparing standard convolution with partial convolution (PConv). Standard convolution operates globally over the input data, whereas partial convolution, incorporating a mask updating mechanism, enables more effective feature extraction and image inpainting when dealing with irregular areas or missing data.

Modeled as follows:

$$x = \begin{cases} W^T(X \odot M\frac{sum(1)}{sum(M)} + b), \ sum(M) > 0 \\ 0, \ otherwise \end{cases} \tag{4}$$

x is the input feature vector, M is the input Mask, W is the convolution kernel, b is the bias. Where $\odot$ denotes element-wise multiplication, and 1 has the same shape as M but with all the elements being 1. As can be seen, the output values depend only on the unmasked inputs. The scaling factor sum(1)/sum(M) applies appropriate scaling to adjust for varying amounts of valid (unmasked) inputs. After each partial convolution operation, we then update our mask as follows: if the convolution was able to condition its output on at least one valid input value, then we mark that location as valid.

PConv achieves rapid and efficient feature extraction by applying filters to only a small subset of the input channels while keeping the remaining channels unchanged. This approach is particularly suitable for running deep learning models on resource-limited devices because it significantly reduces computational demands without sacrificing performance.

3.3. Spatial-to-Depth Convolution

In YOLOv8, the backbone network typically employs pooling for downsampling operations to compress the feature information when processing the input image features. However, this can lead to the loss of subtle feature details and a reduction in effective features, ultimately affecting the model's performance in detecting smaller objects or tasks with lower resolution. To address the issue of easily missed detections when the model detects small target defects, this paper proposes the use of the SPD-Conv module to replace the downsampling module in YOLOv8, as illustrated in Figure 5. The SPD-Conv module is capable of preserving learnable feature information while performing downsampling operations, thereby enhancing the model's detection accuracy for small targets.

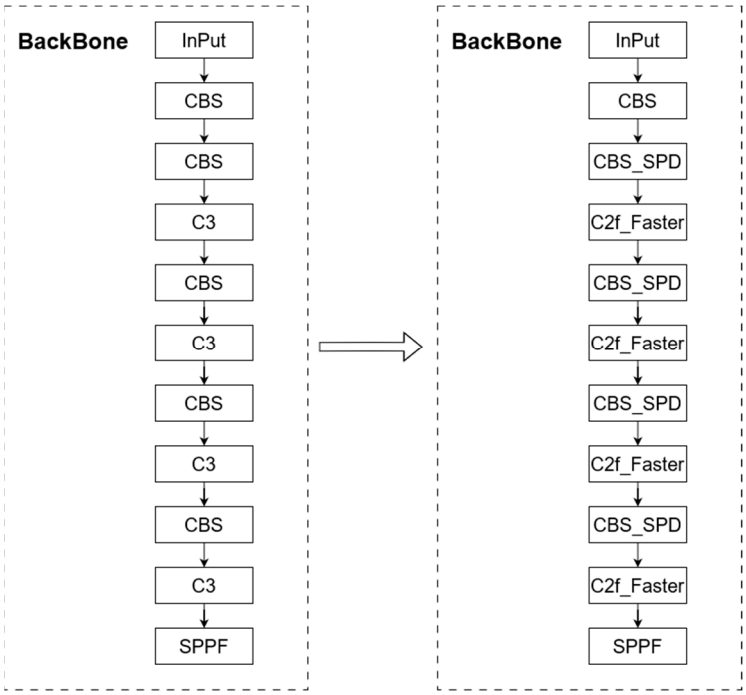

Figure 5. Example of replacing the backbone network in YOLOv8 with SPD-Conv modules. This modification aims to enhance the model's capability to detect low-resolution images and small target defects.

3.3.1. SPD-Conv Network Structure

SPD-Conv represents a novel module within Convolutional Neural Networks (CNNs) that addresses the challenges encountered by traditional CNN models when processing low-resolution images and small objects. By introducing spatial-to-depth transformations and non-striding convolution operations, SPD-Conv successfully enhances the performance of these models in such tasks. Composed of a spatial-to-depth (SPD) layer and a non-striding convolution (Conv) layer, this module can be seamlessly integrated into most CNN architectures. SPD-Conv emerges as an efficient and practical neural network component, and its overall structure is illustrated in Figure 6. The integration of SPD-Conv into CNNs offers a promising approach to improve the detection accuracy and robustness of models dealing with complex visual scenes, particularly those involving low-resolution inputs or the need for precise localization of small objects.

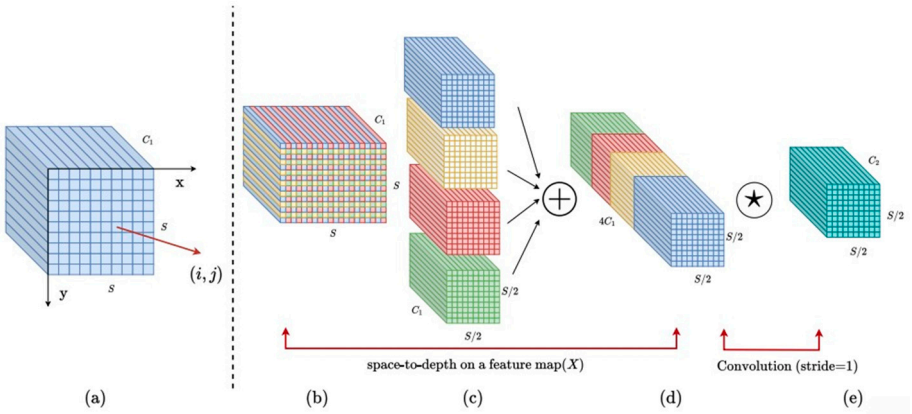

Figure 6. SPD-Conv Network Architecture. By integrating Spatial to Depth (SPD) layers with non-strided convolutional layers, this architecture optimizes the convolutional neural network's handling of low-resolution images and small target defects, enhancing model performance while minimizing

information loss. The key elements of the SPD convolution are composed of (**a**–**e**), Feature map (**a**): A traditional feature map that contains information on the number of channels, height, and width. Space-to-depth transformation (**b**): Through the space-to-depth operation, spatial blocks of pixels are rearranged into the depth/channel dimension, increasing the number of channels to 4 while reducing the spatial dimensions by half. Channel merging (**c**): Different channel groups are merged along the channel dimension. Addition operation (**d**): The merged feature map may be added to other processed feature maps (not detailed in the figure). Stride-1 convolution (**e**): A stride-1 convolution is applied to the resulting feature map, reducing the channel dimension while maintaining spatial resolution, which remains half the original size.

3.3.2. Space to Depth

The spatial-to-depth layer serves the purpose of reducing the spatial dimensions of the input feature maps to the channel dimension, performing downsampling both internally within the CNN and across the entire feature mappings of the CNN, as demonstrated in Equation (5). Through this transformation, information within the channels is preserved while reducing the redundancy in spatial dimensions. Low-resolution images often suffer from issues such as blurriness and distortion, and this conversion assists the model in better handling these images, thereby enabling a more efficient utilization of spatial information.

Consider an intermediate feature map of arbitrary size, which can be partitioned into a series of sub-feature maps as follows:

$$\begin{cases} \begin{cases} f_{0,0} = X[0:S:scale, 0:S:scale], \\ f_{1,0} = X[1:S:scale, 0:S:scale], \\ \cdots \\ f_{scale-1,0} = X[scale-1:S:scale, 0:S:scale]; \\ f_{0,1} = X[0:S:scale, 1:S:scale], f_{1,1} \\ \cdots \\ f_{scale-1,1} = X[scale-1:S:scale, 1:S:scale]; \\ \cdots \\ f_{0,scale-1} = X[0:S:scale-1:S:scale], f_{1,scale-1,}, \\ \cdots \\ f_{scale-1,scale-1} = X[scale-1:S:scale, sclae-1:S:scale]. \end{cases} \end{cases} \tag{5}$$

X is the intermediate feature map; its dimension is $S \times S \times C_1$, $f_{0,0}f_{1,0}f_{0,1}f_{1,1}$ respectively the four subgraphs obtained in Figure 6c.

The non-striding convolution layer is applied subsequent to the SPD layer, performing conventional convolution operations without including a stride. This implies that during convolution, the input and output feature maps maintain identical spatial dimensions. This design choice facilitates the preservation of fine-grained information, preventing the loss of crucial features during the convolution process.

3.3.3. Non-Stride Convolutional Layer

The non-striding convolution layer represents a convolution operation that excludes the use of a stride (i.e., with a stride of 1) during the convolution process, thus eliminating the need to move across the feature map. This operation contributes to the preservation of finer-grained information, enabling feature extraction without reducing the size of the feature map. In traditional convolutional neural network (CNN) architectures, as the network layers deepen, the spatial resolution of images gradually decreases when stride convolutions and pooling layers are directly applied, leading to the loss of detailed information for small objects. The non-striding convolution layer circumvents this issue, playing a crucial role in improving the recognition performance for low-resolution images and small objects.

The non-striding convolution layer is often combined with the spatial-to-depth layer to form building blocks such as SPD-Conv. SPD-Conv enhances the model's detection

capabilities for low-resolution images and small objects, reducing its dependence on "good-quality" inputs. Consequently, the non-striding convolution layer occupies a significant role in tasks such as object detection.

3.4. Weighted Bidirectional Feature Pyramid Network (BiFPN)

The Feature Pyramid is a widely used technique in computer vision, particularly for tasks such as object detection, image segmentation, and target tracking. Its main idea is to extract features from different image scales, capturing information about objects of varying sizes and resolutions. YOLOv8 follows the PAN concept introduced in YOLOv5, adopting PAN-FPN as its feature extraction network. PAN-FPN incorporates both top- down and bottom-up paths, preserving more spatial information and thereby enhancing the accuracy and robustness of model training.

However, when addressing the problem of small target defects in the packaging of bags on industrial assembly systems, the PAN-FPN network exhibits high computational complexity and insufficient spatial feature fusion capabilities. This reduces the efficiency of utilizing shallow features and fails to meet the requirements for detecting small target defects. Therefore, this paper introduces a weighted bidirectional feature pyramid network (BiFPN) to replace the original feature pyramid network. BiFPN enhances the network's ability to extract shallow features while reducing computational complexity, thereby improving the overall performance of the model in detecting small target defects.

3.4.1. BiFPN Network Structure

The bidirectional feature pyramid network (BiFPN) is a multilevel feature pyramid network designed to address the issues of feature propagation and information flow in object detection. Building upon the traditional feature pyramid network (FPN), BiFPN introduces mechanisms such as bidirectional feature fusion and feature selection to enhance the efficiency and accuracy of feature extraction. The network structure is weighted and bidirectionally connected, encompassing both top-down and bottom-up paths. By establishing bidirectional channels for cross-scale connections, features from the feature extraction network are directly fused with corresponding-sized features from the bottom-up path, preserving shallower semantic information without sacrificing too much deep semantic information. Figure 7 compares the network structures of FPN, PANet, and BiFPN.

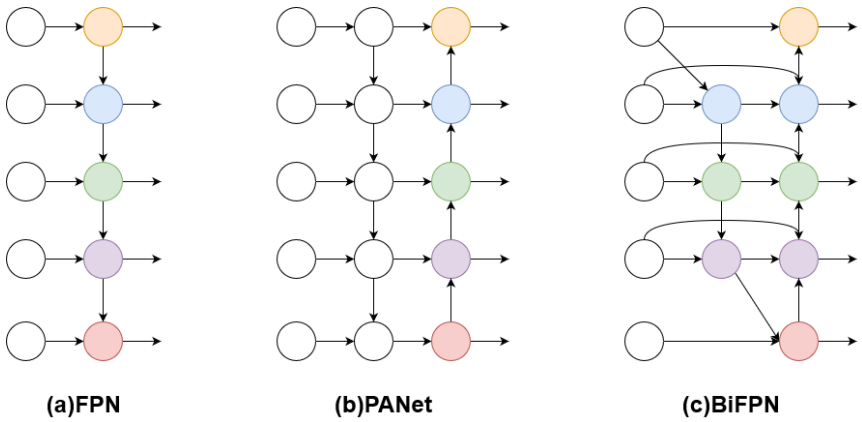

Figure 7. (**a**) FPN Network Structure. (**b**) PANet Network Structure, which proposes a bidirectional fusion backbone network based on FPN, with a top-down and bottom-up pathway. (**c**) BiFPN Network Structure, which treats each bidirectional (top-down and bottom-up) path as a feature network layer.

3.4.2. Cross-Scale Connections

Cross-scale connections refer to the establishment of bidirectional channels that enable information flow and fusion between features of different scales. This connection allows for more effective flow and fusion of information between features of different scales by

establishing bidirectional connections between the top-down and bottom-up paths. BiFPN optimizes upon the feature pyramid network (FPN) by introducing learnable weights to learn the importance of different input features and repeatedly applying multiscale feature fusion in both top-down and bottom-up directions. This cross-scale connection helps preserve shallower semantic information without sacrificing too much deep semantic information, thereby enhancing the accuracy and efficiency of feature extraction.

In BiFPN, this is primarily reflected in the following aspects: Firstly, nodes with only one input edge are removed as they lack feature fusion and contribute minimally to network fusion. Deleting these nodes has minimal impact on the network while simplifying the bidirectional structure. Secondly, additional edges are added between the original input and output nodes at the same level to fuse more features without incurring significant computational costs. Finally, to achieve higher-level feature fusion, each bidirectional path is treated as a feature network layer and repeated multiple times to achieve more advanced feature fusion, enhancing the richness and accuracy of features, as shown in Figure 8.

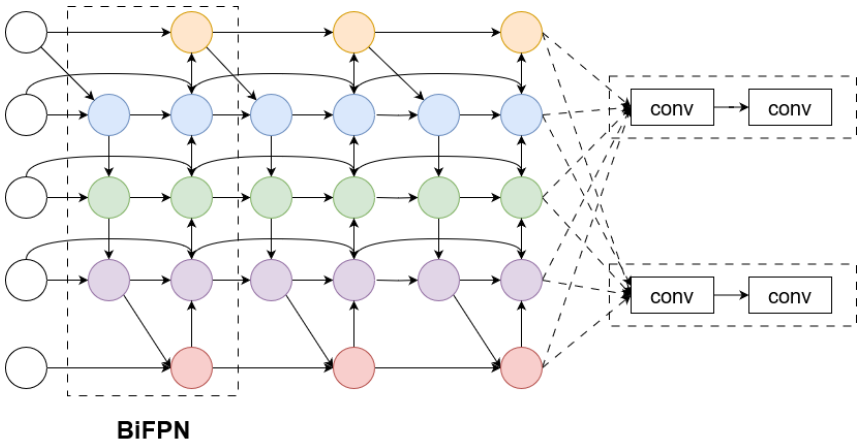

Figure 8. Repeated Bidirectional Feature Pyramid Network (BiFPN) Structure for Advanced Feature Fusion. In the BiFPN layer, we can observe how features at different scales are fused through bidirectional paths. This structural design aims to maximize the effectiveness of feature fusion while maintaining computational efficiency, thereby enhancing the overall performance of object detection. The diagram also shows the classification network and bounding box regression network, which are parts of the network used to predict object classes and locate object bounding boxes after BiFPN feature fusion.

Traditional feature fusion methods typically involve operations such as concatenation or addition to combine input features (feature maps) without effectively distinguishing between them. However, different feature maps contribute differently to feature fusion, and simply adding or concatenating them for fusion is not the most ideal operation. Therefore, BiFPN proposes an efficient weighted feature fusion mechanism.

Modeled as follows:

$$x = \begin{cases} P_i^{td} = Conv\left(\frac{w_1 \cdot P_i^{in} + w_2 \cdot Resize\left(P_{i+1}^{in}\right)}{w_1 + w_2 + \varepsilon}\right) \\ P_i^{out} = Conv\left(\frac{w_1' \cdot P_i^{in} + w_2' \cdot P_i^{td} + w_3' \cdot Resize\left(P_{i-1}^{out}\right)}{w_1' + w_2' + w_3' + \varepsilon}\right) \end{cases} \quad (6)$$

p is the Feature Map, *Resize* is the upsampling or downsampling operation, w is the learning parameter used to distinguish the importance of different features in the feature fusion process.

3.5. Improve the Loss Function

The YOLO series of models has derived various loss functions based on IoU for calculating losses. Both the YOLOv5 and YOLOv8 models employ the CIoU function, as shown

in Equation (7). IoU is used to assess the overlap between the predicted bounding box (the target location predicted by the model) and the ground-truth box (the actual target location). This metric reflects the detection performance between the predicted and ground-truth boxes, where a higher overlap indicates better detection. However, this IoU measurement can be highly sensitive to minor positional offsets for small targets. For tiny objects, their pixel values are relatively small, and even slight shifts in the predicted bounding box may result in an IoU below the set threshold, causing them to be falsely classified as negative samples. This can hinder the effective convergence of the network model.

Modeled as follows:

$$l_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{7}$$

IoU is the intersection ratio, b is the center point of the rectangular frame, ρ is the Euclidean distance between the two rectangular frames, c is the diagonal distance of the closed areas of the two rectangular frames, v is used to measure the consistency of the relative proportions of the two rectangular frames, and α is the weight coefficient.

In a practical industrial product packaging bag assembly line, defects such as packaging damage and black spots caused by upstream equipment only account for a small portion of the surface of the packaging, and their characteristics can easily be confused with the information on the packaging surface. The existence of small target defects like packaging damage and black spots increases the difficulty of extracting network features. In such scenarios, the IoU loss fails to optimize the loss effectively, resulting in small gradients of the function and poor network convergence. Therefore, this paper introduces the NWD loss function as a measurement method for small targets, which can effectively reduce the sensitivity to positional deviations of small target defects.

3.5.1. NWD Loss Function

The NWD loss function is a similarity metric method that does not rely on the overlap of the bounding boxes. Its main characteristics include the introduction of sample weights and normalization to better handle the differences in the importance of different samples and measure the similarity between samples. In the process of defect detection, the information of the target object and the background information are concentrated in the central part and boundary part of the bounding box, respectively. Firstly, the bounding box is modeled as a 2D Gaussian distribution, with the weight of the central pixel set to the maximum value and decreasing gradually from the center to the boundary. Secondly, a new metric called the Normalized Wasserstein Distance (NWD) is proposed, which helps to more accurately measure the similarity between two Gaussian distributions. Specifically, when detecting small targets, they occupy only a few pixels in size and usually lack significant information, making it difficult for traditional object detectors to achieve satisfactory results. Therefore, the introduction of the Wasserstein distance allows for the calculation of the similarity between their corresponding Gaussian distributions, thus reducing the sensitivity of IoU to positional deviations of small target defects. The NWD loss function is represented by Equation (8).

Modeled as follows:

$$L_{NWD} = 1 - NWD(N_p, N_g) \tag{8}$$

L_{NWD} is It is used to guide model training. The goal is to make the NWD between the predicted Gaussian distribution N_P and the real Gaussian distribution N_g as small as possible; that is, the closer they are, the smaller the loss and the better the model performance.

3.5.2. Bounding Box Gaussian Distribution Modeling

Small target defects, such as damage and black spots, are mostly not strictly rectangular, and their bounding boxes often contain some background pixels. The foreground pixels and background pixels are concentrated in the center and boundaries of the bounding box, respectively. To better describe the weights of different pixels within the bounding box,

the bounding box is modeled as a two-dimensional (2D) Gaussian distribution, where the central pixel of the bounding box has the highest weight and the pixel weights decrease gradually from the center to the boundary. For a horizontal bounding box, where, and represent the center coordinates, width, and height, respectively, its formula is shown in Equation (9).

Modeled as follows:

$$\frac{(x-\mu_x)^2}{\sigma_x^2}+\frac{(y-\mu_y)^2}{\sigma_y^2}=1 \tag{9}$$

(μ_x, μ_y) is the center coordinate of the ellipse, (σ_x, σ_y) is the length of the semi-axis along the x and y axes.

The probability density function of the two-dimensional Gaussian distribution is shown in Equation (10):

Modeled as follows:

$$f(x|\mu,\sum)=\frac{\exp\left(-\frac{1}{2}(x-\mu)^{T\sum^{-1}}(x-\mu)\right)}{2\pi|\sum|^{\frac{1}{2}}} \tag{10}$$

x represents the Gaussian distribution coordinates, μ represents the mean vector, $\sum$ represents the covariance matrix.

When these conditions are satisfied, the horizontal bounding box can be modeled as a two-dimensional Gaussian distribution, as shown in Equation (11).

Modeled as follows:

$$\mu=\begin{bmatrix}c_x\\c_y\end{bmatrix},\sum=\begin{bmatrix}\frac{w^2}{4} & 0\\0 & \frac{w^2}{4}\end{bmatrix} \tag{11}$$

3.5.3. Normalized Gaussian Wasserstein Distance

By calculating the Wasserstein distance between the corresponding Gaussian distributions, we can more accurately assess the similarity between them. This metric can be easily embedded into the label assignment, non-maximum suppression, and loss functions of anchor-based detectors, replacing the commonly used IoU metric and thereby improving the accuracy of small object detection. The second-order formula for this metric is shown in Equation (12).

Modeled as follows:

$$W_2^2(N_a,N_b)=||([cx_a,cy_a,\frac{w_a}{2},\frac{h_a}{2}]^T,[cx_b,cy_b,\frac{w_b}{2},\frac{h_b}{2}]^T)||_2^2 \tag{12}$$

$W_2^2(N_a, N_b)$ represents the square of the second-order Wasserstein distance between Gaussian distributions N_a and N_b

However, this formula cannot be used directly for similarity measurements. Instead, a normalized exponent needs to be applied to obtain a new metric known as the Normalized Wasserstein Distance (NWD), as shown in Equation (13).

$$(N_a,N_b)=\exp(-\frac{\sqrt{W_2^2(N_a,N_b)}}{C}) \tag{13}$$

4. Experimental Results

In this section, the defect detection performance of the improved YOLOV8 algorithm for subtle defects in industrial product packaging bags is systematically analyzed and evaluated. The detection results of the proposed algorithm for various subtle defects are summarized. To validate the detection performance of the FSNB_YOLOV8 network structure in scenarios such as fine black spots and damage, trained weights are first used to verify the model. Evaluation metrics, including accuracy (Equation (14)), recall (Equation (15)), mean average precision (Equations (16) and (17)), and FPS are employed to assess the

model's performance. Precision refers to the proportion of truly positive samples among the outcomes predicted by the model as positive and measures the accuracy of the model's predictions for positive samples. Recall represents the proportion of actual positive samples that are correctly predicted as positive by the model, evaluating the model's ability to identify all positive samples. The area under the precision-recall curve is the average precision (AP) for that particular category. Subsequently, to explore the impact of each module on the model, ablation experiments are conducted for each improvement item using the YOLOv8s model as the baseline. The same training data and parameter settings are applied for these experiments. Finally, comparisons are made with existing detection methods, including Faster R-CNN, SSD, RetinaNet, YOLOv5, and YOLOV8. All the aforementioned experiments are conducted on a system equipped with an NVIDIA GeForce GTX 4090.

Modeled as follows:

$$P = \frac{TP}{TP + FP} \times 100\% \tag{14}$$

$$R = \frac{TP}{TP + FN} \times 100\% \tag{15}$$

P represents accuracy, R represents recall rate, TP indicates marking positive samples as positive samples, FP indicates marking negative samples as positive samples, FN indicates marking positive samples as negative samples, TN indicates labeling negative samples as negative samples.

$$AP = \int_0^1 P(R)dR \times 100\% \tag{16}$$

$$mAP = \frac{1}{c}\sum_{i=1}^{c} AP_i = \frac{1}{C}\int_0^1 P(R)dR \tag{17}$$

c is the number of categories, AP is the average accuracy, mAP is the mean average accuracy.

4.1. Experimental Parameters

The specific parameters of the experimental environment configuration are shown in Table 1. The specific parameters of experimental model training are shown in Table 2.

Table 1. Environment configuration and parameters.

Configuration	Type
OS	Windows10
CPU	i9-11900k
GPU	4090
CUDA	11.8
Python	3.10
PyTorch	2.0.1
Visual Studio	2019

Table 2. Training parameter settings.

Parameters	Value
Batch Size	32
Lr	0.01
Mixup	0.5
Mosaic	0.8
Copy_Paste	0.5
Epoch	300
Momentum	0.937

4.2. Experimental Data

The experimental dataset consists of real-world samples collected from an actual industrial production line. The data is carefully screened, including 400 defective samples and 4000 normal samples. The defects encompass four classical types: black spots, scratches, tears, and indentations, with 100 samples for each type. Utilizing an introduced data augmentation algorithm, each defective sample is expanded to 1000 images. Additionally, 400 normal images without defects are added as background inputs during model training to reduce the false positive rate. The dataset is randomly split into training, testing, and validation sets in an 8:1:1 ratio.

4.3. Experimental Results

The accuracy curve of the improved FSNB_YOLOv8s is presented in Figure 9, the precision-recall (P-R) curve is depicted in Figure 10, and the detection result illustrations are shown in Figure 11. Among them, 'afdamage' represents tear defects, and 'heidian' stands for black spot defects. The improved mAP achieves 98.8%, and the model training metrics are displayed in Figure 12.

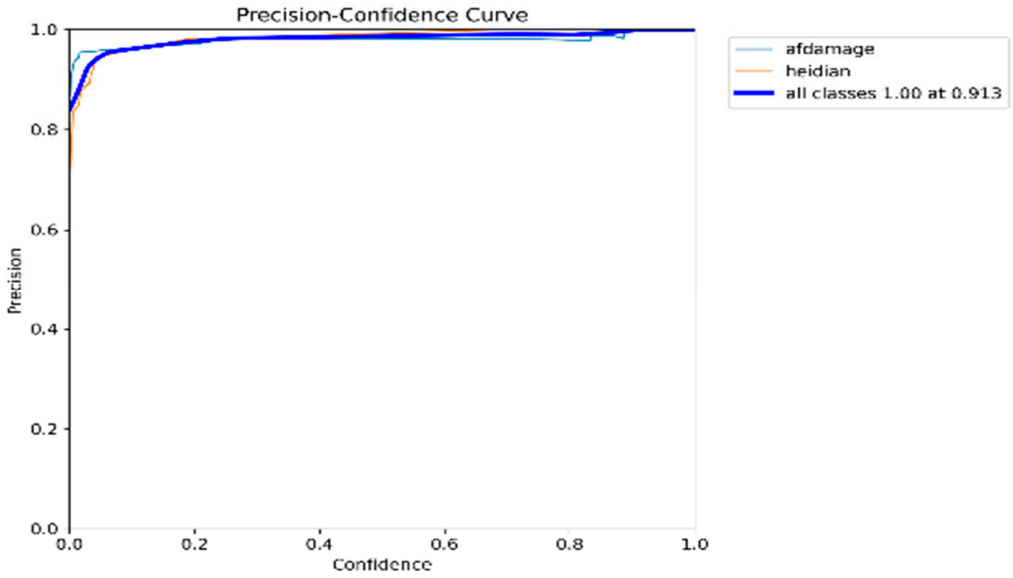

Figure 9. Diagram of accuracy for detecting black spots and aluminum foil tears.

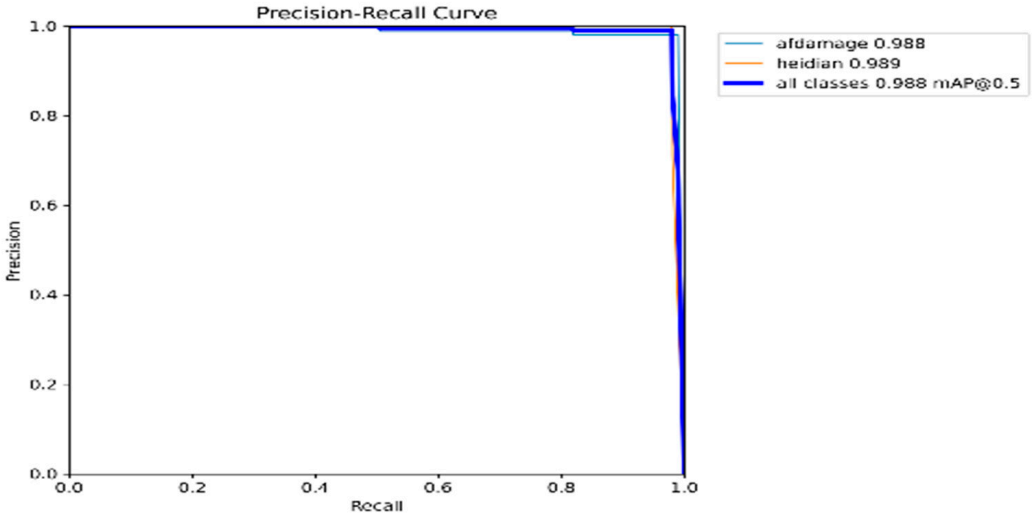

Figure 10. Diagram of Precision-Recall (P-R) Curve for Detecting Black Spots and Aluminum Foil Tears.

Figure 11. FSNB_YOLOv8 Defect Detection Result Example. The Chinese characters on the surface of the medicine bag indicate the name of the medicine bag particles and the specific packaging requirements. From the above picture, we can see that small defects can be identified by the model.

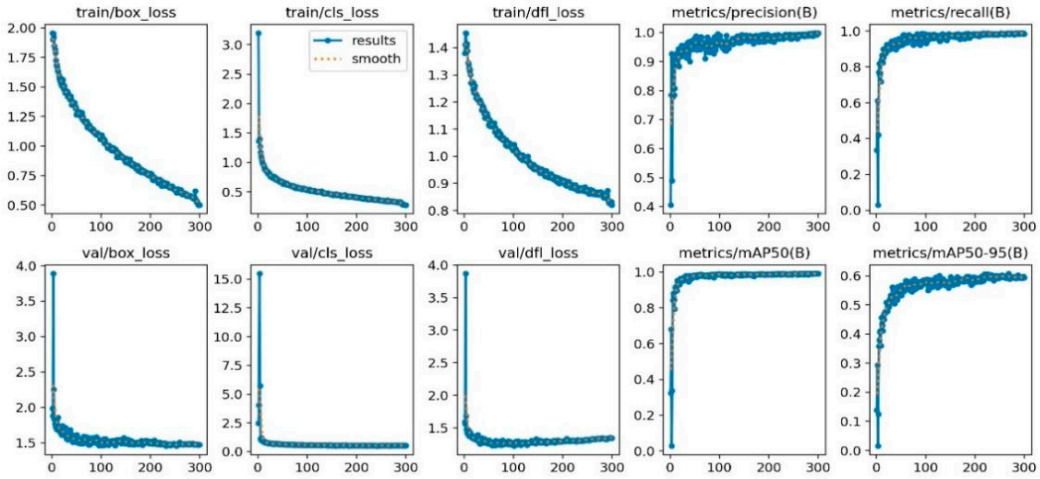

Figure 12. FSNB_YOLOv8 Model Training Result Example, the x-axis represents the number of training iterations.

4.4. Ablation Experiment

In order to explore the impact of each module on the model, based on the YOLOv8s model, the experiment used the same training data and parameter settings, and conducted ablation experiments for each improvement item. The effects are shown in Table 3.

Table 3. Impact of different improvement methods on the performance of the YOLOv8 algorithm.

	FasterNet (F)	SPD (S)	BiFPN (B)	NWD (N)	mAP (%)	Precision (%)	Recall (%)	FPS
YOLOv8					92.6	96.5	91.7	65
YOLOv8+F	√				92.8	96.8	92.1	76
YOLOV8+FS	√	√			96.2	98.5	94.4	73
YOLOV8+FSB	√	√	√		97.3	98.8	95.2	71
YOLOV8+FSBN	√	√	√	√	98.8	99.2	97.6	74

As can be seen from Table 3, the replaced modules in the network all contribute to the improvement in model performance, with the SPD-Conv module exhibiting the most significant enhancement in mean average precision (mAP), achieving a 3.4% increase. The SPD-Conv module preserves the feature information lost due to previous downsampling, making it easier for the model to extract feature information for small targets, thereby enhanc-

ing the accuracy of detecting small defect targets. By replacing the FasterNet module, the FPS of the model increases by 11FPS compared to the original model, effectively improving the inference speed of the model. Furthermore, the introduction of the NWD loss function also leads to a notable improvement in mAP, with a 1.5% enhancement. The NWD module significantly reduces the sensitivity to the positional deviation of small defect targets, enabling more accurate identification of minute defects. By integrating the FasterNet, SPD-Conv, BiFPN, and NWD modules, the final model achieves an mAP of 98.8%, with accuracy and recall rates reaching 99.2% and 97.6%, respectively, and a model inference speed of 74FPS.

To visually demonstrate the detection performance of the improved algorithm, this study selected a few representative images from the test dataset, as shown in Figures 13 and 14. Figure 13 displays the detection results using the YOLOv8s model, and Figure 14 presents the detection outcomes using the FSNB_YOLOv8s model. Through comparison, it can be observed that the FSNB_YOLOv8s model is capable of detecting finer tear and black spot defects and performs better than the original model in terms of defect localization and confidence prediction.

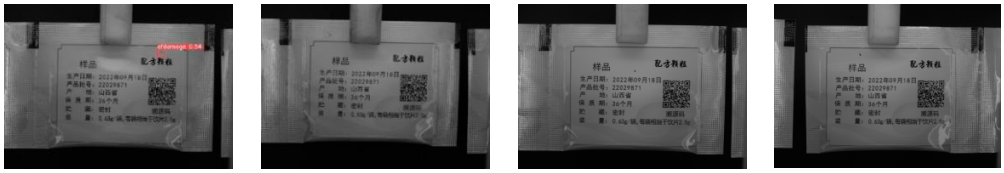

Figure 13. The defect detection results of the original YOLOv8s model. The Chinese characters on the surface of the medicine bag indicate the name of the medicine bag particles and the specific packaging requirements. From the above picture, we can see that the small defects cannot be identified by the original model.

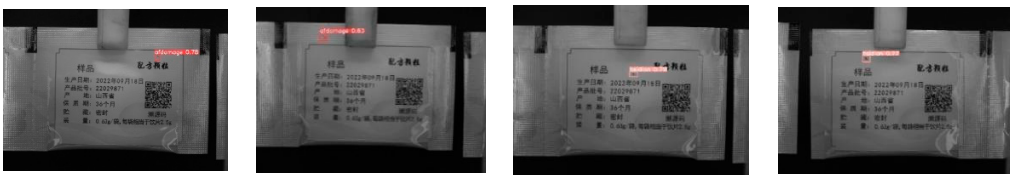

Figure 14. The defect detection results of the FSNB_YOLOv8s model. The Chinese characters on the surface of the medicine bag indicate the name of the medicine bag particles and the specific packaging requirements. From the above picture, we can see that small defects can be identified by the model.

4.5. Comparative Experiment

The proposed algorithm model, Faster R-CNN, CenterNet, YOLOv4, and YOLOv5 were trained and tested on the same dataset, and the results are summarized in Table 4. Faster R-CNN, CenterNet, YOLOv4, and YOLOv5 did not perform ideally in the detection accuracy of subtle defects. The proposed algorithm model effectively improves the accuracy of identifying fine defects, such as tiny black spots and tears on the surface of packaging, by employing various strategies, including designing data enhancement algorithms, preserving defect feature information, performing efficient feature fusion, and reducing the sensitivity to positional deviations. Additionally, it effectively reduces the training time and inference time of the model without compromising defect feature extraction. Finally, the effectiveness of the proposed network is demonstrated.

Table 4. Performance comparison of other models.

	FPS	**mAP (%)**
FasterR-CNN	5	81.3
CenterNet	28	79.8
YOLOv4	40	88.6
YOLOv5	56	91.4
FSNB_YOLOv8	74	98.8

4.6. Discussion

In industrial production environments, the detection of surface defects in industrial product packaging bags is a vital quality control task. However, this endeavor confronts multiple challenges, the chief of which is the marked data imbalance stemming from an overwhelming abundance of samples representing normal packaging contrasted with a dearth of those displaying defects, resulting in a pronounced long-tailed distribution. Such a skewed distribution can predispose deep learning models to overemphasize common patterns during training at the expense of learning rare defect modes, thereby impeding their generalization capabilities. Additionally, the computationally intensive and parameter-heavy C2F module in the YOLOv8 model renders it inefficient and unsuitable for real-time deployment on edge devices with limited computational resources, contravening the practical requirements of industrial settings.

In response to the aforementioned challenges, the FSNB_YOLOv8 model presents a suite of innovative solutions. Firstly, it employs an interactive data augmentation algorithm based on fuzzy search, which proactively addresses data skewness. This approach intelligently generates and enhances defect samples, expanding the scale of the defect dataset and thereby mitigating data imbalance issues. It enables the model to thoroughly learn diverse defect patterns during training, enhancing its ability to recognize rare defects in actual production scenarios and bolstering the overall generalization performance.

Secondly, FSNB_YOLOv8 substitutes the backbone network of YOLOv8 with a composite network comprising FasterNet and SPD-Conv, effectively managing computational complexity and network redundancy. This design aims to maintain detection accuracy while significantly reducing the computational cost of the model, rendering it more compatible with the computational constraints of edge devices, and facilitating real-time, efficient deployment in situ.

Additionally, the model harnesses a Bidirectional Feature Pyramid Network (BiFPN) to enhance its capability to detect defects across various scales. BiFPN adeptly integrates feature information from different hierarchical levels, ensuring that when confronted with defects of varying sizes, the model can capture crucial features, thereby enhancing detection precision and robustness.

Lastly, the adoption of the NWD loss function improves the model's sensitivity to positional deviations in defects. Compared to conventional loss functions, the NWD loss function more finely quantifies the distance discrepancies between the predicted and ground-truth bounding boxes, enabling the model to more accurately localize defect positions, thereby reducing misclassifications due to localization errors and ultimately augmenting detection accuracy.

5. Conclusions

The FSNB_YOLOv8 model addresses the unique challenges inherent in pouch packaging surface defect detection within industrial production systems through a range of strategies, encompassing data augmentation and balancing, network architecture optimization, multiscale feature fusion, and localization performance refinement. These measures collectively constitute effective enhancements to the original YOLOv8 model. Experimental results attest that the FSNB_YOLOv8 model exhibits a 3.1 percentage point improvement in the mAP50 metric relative to CL_YOLOv5, substantiating the efficacy of these optimization efforts. Consequently, the FSNB_YOLOv8 model not only successfully resolves critical issues such as data imbalance, computational resource constraints, and precision in detecting small-scale targets but also furnishes an industrially applicable solution for the detection of surface defects on industrial products packaging pouches in production systems imbued with both theoretical significance and practical relevance.

Author Contributions: Investigation, Software, Writing Original Draft, Validation, Methodology, J.W.; Conceptualization, Methodology, Supervision, and Writing—Review and Editing, J.L.; computing

resources and automated data collection, J.W.; data curation and writing, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Basic Research Project of the Translational Application Project of the "Wise Eyes Action" (Project No. F2B6A194). We would like to express our deepest gratitude to these organizations for their generous funding and support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: We would like to express our gratitude to Yanhua Shao (The Sixth Research Institute of China Electronics and Information Industry Corporation) for providing the computational resources and support. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ma, B.; Li, Q. High Speed Pharmaceutical Packaging Detection System Based on Genetic Algorithm and Memory Optimization. In Proceedings of the International Conference on Cloud Computing and Security, Nicosia, Cyprus, 10–13 December 2018; Springer: Cham, Switzerland, 2018; pp. 356–368.
2. He, Y.; Song, K.; Meng, Q.; Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [CrossRef]
3. Zhao, Z.B.; Wang, C.X.; Zhang, X.W.; Chen, X.F.; Li, Y.H. Research progress and challenges in process intelligent monitoring of laser powder bed fusion additive manufacturing. *Engineering* **2023**, *59*, 253–276.
4. Fu, X.; Yang, X.; Zhang, N.; Zhang, R.; Zhang, Z.; Jin, A.; Ye, R.; Zhang, H. Bearing surface defect detection based on improved convolutional neural network. *Math. Biosci. Eng.* **2023**, *20*, 12341–12359. [CrossRef]
5. Jun, L.; Jiajie, Z.; Jinglei, W.; Yanzhao, L. Improved Drug Traceability Code Detection Algorithm Based on YOLOv5. *Proc. J. Phys. Conf. Ser.* **2023**, *2589*, 012003. [CrossRef]
6. Sun, Y.; Chen, Y.; Wu, P.; Wang, X.; Wang, Q. DRL: Dynamic rebalance learning for adversarial robustness of UAV with long-tailed distribution. *Comput. Commun.* **2023**, *205*, 14–23. [CrossRef]
7. Chu, P.; Bian, X.; Liu, S.; Ling, H. Feature space augmentation for long-tailed data. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXIX 16. Springer International Publishing: Cham, Switzerland, 2020; pp. 694–710.
8. Shakir, S.; Topal, C. Unsupervised fabric defect detection with local spectra refinement (LSR). *Neural Comput. Appl.* **2024**, *36*, 1091–1103. [CrossRef]
9. Birla, S.; Alya, S.; Singh, R. An integrated image processing approach for 3D scanning and micro-defect detection. *J. Micromanuf.* **2023**, *6*, 172–181. [CrossRef]
10. Davidson, J.L. Classification of Lattice Transformations in Image Processing. *Comput. Vis. Image Underst.* **1993**, *57*, 283–306. [CrossRef]
11. Shutaro, Y. Image Processing Device, Control Method, and Program. WO2018088238A1, 17 May 2018.
12. Li, D.; Wu, H.; Zhang, J.; Huang, K. Fast a3rl: Aesthetics-aware adversarial reinforcement learning for image cropping. *IEEE Trans. Image Process.* **2019**, *28*, 5105–5120. [CrossRef]
13. He, Y.; Ye, Y.; Hanhart, P.; Xiu, X. Motion compensated prediction with geometry padding for 360 video coding. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; IEEE: New York, NY, USA, 2017; pp. 1–4.
14. Cruz-Roa, A.A.; Arevalo Ovalle, J.E.; Madabhushi, A.; González Osorio, F.A. A deep learning architecture for image representation, visual interpretability and automated basal-cell carcinoma cancer detection. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013: 16th International Conference, Nagoya, Japan, 22–26 September 2013; Proceedings, Part II 16. Springer: Berlin/Heidelberg, Germany, 2013; pp. 403–410.
15. Jozef, G.; Cassano, J.; Dahlke, S.; de Boer, G. Testing the efficacy of atmospheric boundary layer height detection algorithms using uncrewed aircraft system data from MOSAiC. *Atmos. Meas. Tech. Discuss.* **2022**, *15*, 4001–4022. [CrossRef]
16. Zou, D.; Cao, Y.; Li, Y.; Gu, Q. The benefits of mixup for feature learning. In Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023; PMLR: New York, NY, USA, 2023; pp. 43423–43479.
17. Huaizhou, Y.; Danyang, X. Research and Analysis of Image Enhancement Algorithm in the Classification of Rock Thin Section Images. In Proceedings of the 2021 3rd International Conference on Intelligent Control, Measurement and Signal Processing and Intelligent Oil Field (ICMSP), Xi'an, China, 23–25 July 2021; IEEE: New York, NY, USA, 2021; pp. 125–128.

18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
19. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6568–6577.
20. 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.
21. Stergiou, A.; Poppe, R. Adapool: Exponential adaptive pooling for information-retaining downsampling. *IEEE Trans. Image Process.* **2022**, *32*, 251–266. [CrossRef] [PubMed]
22. Cheng, W.; Liang, P.; Cao, S. Texture Detect on Rotary-Veneer Surface Based on Semi-Fuzzy Clustering Algorithm. In Proceedings of the Computer and Computing Technologies in Agriculture III: Third IFIP TC 12 International Conference, CCTA 2009, Beijing, China, 14–17 October 2009; Revised Selected Papers 3. Springer: Berlin/Heidelberg, Germany, 2010; pp. 369–374.
23. Zhang, W.; Zeng, Y.; Wang, J.; Ma, H.; Zhang, Q.; Fan, S. Multi-scale feature pyramid approach for melt track classification in laser powder bed fusion via coaxial high-speed imaging. *Comput. Ind.* **2023**, *151*, 103975. [CrossRef]
24. Zhan, C.; Duan, X.; Xu, S.; Song, Z.; Luo, M. An improved UAV object detection algorithm based on ASFF-YOLOv5s. *Math. Biosci. Eng. MBE* **2023**, *20*, 10773–10789.
25. Zhou, S.; Zhao, J.; Shi, Y.S.; Wang, Y.F.; Mei, S.Q. Research on improving YOLOv5s algorithm for fabric defect detection. *Int. J. Cloth. Sci. Technol.* **2023**, *35*, 88–106. [CrossRef]
26. Bawankule, R.; Gaikwad, V.; Kulkarni, I.; Kulkarni, S.; Jadhav, A.; Ranjan, N. Visual Detection of Waste using YOLOv8. In Proceedings of the 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 14–16 June 2023; pp. 869–873.
27. Hobert, J.P. The data augmentation algorithm: Theory and methodology. In *Handbook of Markov Chain Monte Carlo*; CRC Press: Boca Raton, FL, USA, 2011; pp. 253–293.
28. Guo, A.; Sun, K.; Zhang, Z. A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection. *J. Real-Time Image Process.* **2024**, *21*, 49. [CrossRef]
29. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; Springer Nature: Cham, Switzerland, 2022; pp. 443–459.
30. 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.
31. Wang, J.; Xu, C.; Yang, W.; Yu, L. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. *arXiv* **2021**, arXiv:2110.13389.
32. Xu, H.; Gao, Y.; Yu, F.; Darrell, T. End-to-end learning of driving models from large-scale video datasets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2174–2182.
33. Feng, J.; Wang, Z.; Wang, S.; Tian, S.; Xu, H. MSDD-YOLOX: An enhanced YOLOX for real-time surface defect detection of oranges by type. *Eur. J. Agron.* **2023**, *149*, 126918. [CrossRef]
34. Ling, Z.; Zhang, A.; Ma, D.; Shi, Y.; Wen, H. Deep Siamese Semantic Segmentation Network for PCB Welding Defect Detection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5006511. [CrossRef]
35. Li, C.; Huang, Y.; Li, H.; Zhang, X. A weak supervision machine vision detection method based on artificial defect simulation. *Knowl.-Based Syst.* **2020**, *208*, 106466. [CrossRef]
36. Liu, J.; Guo, F.; Gao, H.; Li, M.; Zhang, Y.; Zhou, H. Defect detection of injection molding products on small datasets using transfer learning. *J. Manuf. Process.* **2021**, *70*, 400–413. [CrossRef]
37. Li, J.; Su, Z.; Geng, J.; Yin, Y. Real-time detection of steel strip surface defects based on improved yolo detection network. *IFAC-Pap.* **2018**, *51*, 76–81. [CrossRef]
38. Liu, J.; Wang, C.; Su, H.; Du, B.; Tao, D. Multistage GAN for Fabric Defect Detection. *IEEE Trans. Image Process.* **2020**, *29*, 3388–3400. [CrossRef] [PubMed]
39. Pan, Y.; Zhang, L. Dual attention deep learning network for automatic steel surface defect segmentation. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**, *37*, 1468–1487. [CrossRef]
40. He, K.; Liu, X.; Liu, J.; Wu, P. A multitask learning-based neural network for defect detection on textured surfaces under weak supervision. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5016914. [CrossRef]