electronics

MDPI

Article

YOLO-Xray: A Bubble Defect Detection Algorithm for Chip X-ray Images Based on Improved YOLOv5

Jie Wang [1,†], Bin Lin [2,†], Gaomin Li [1], Yuezheng Zhou [1], Lijun Zhong [1], Xuan Li [3] and Xiaohu Zhang [1,*]

[1] School of Aeronautics and Astronautics, Sun Yat-sen Unviersity, Guangzhou 510725, China; wangj688@mail2.sysu.edu.cn (J.W.); zhonglj9@mail.sysu.edu.cn (L.Z.)
[2] College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350117, China; linb@fjnu.edu.cn
[3] China Aerospace Components Engineering Center, Beijing 100094, China
* Correspondence: zhangxiaohu@mail.sysu.edu.cn
† These authors contributed equally to this work.

Abstract: In the manufacturing of chips, the accurate and effective detection of internal bubble defects of chips is essential to maintain product reliability. In general, the inspection is performed manually by viewing X-ray images, which is time-consuming and less reliable. To solve the above problems, an improved bubble defect detection model YOLO-Xray based on the YOLOv5 algorithm for chip X-ray images is proposed. First, the chip X-ray images are preprocessed by image segmentation to construct the chip X-ray defect dataset, namely, CXray. Then, in the input stage, the K-means++ algorithm is used to re-cluster the CXray dataset to generate the anchors suitable for our dataset. In the backbone network, a micro-scale detection head is added to improve the capabilities for small defect detection. In the neck network, the bi-direction feature fusion idea of BiFPN is used to construct a new feature fusion network based on the improved backbone to fuse the semantic features of different layers. In addition, the Quality Focal Loss function is used to replace the cross-entropy loss function to solve the imbalance of positive and negative samples. The experimental results show that the mean average precision (mAP) of the YOLO-Xray algorithm on the CXray dataset reaches 93.5%, which is 5.1% higher than the original YOLOv5. Meanwhile, the YOLO-Xray algorithm achieves state-of-the-art detection accuracy and speed compared with other mainstream object detection models. This shows the proposed YOLO-Xray algorithm can provide technical support for bubble defect detection in chip X-ray images. The CXray dataset is also open and available at CXray.

Keywords: bi-direction feature fusion; bubble defect detection; deep learning; X-ray images; YOLOv5

Citation: Wang, J.; Lin, B.; Li, G.; Zhou, Y.; Zhong, L.; Li, X.; Zhang, X. YOLO-Xray: A Bubble Defect Detection Algorithm for Chip X-ray Images Based on Improved YOLOv5. *Electronics* **2023**, *12*, 3060. https://doi.org/10.3390/electronics12143060

Academic Editors: Teng Huang, Qiong Wang and Yan Pang

Received: 25 June 2023
Revised: 8 July 2023
Accepted: 10 July 2023
Published: 13 July 2023

1. Introduction

With the widespread use of semiconductor chips across a wide field of applications, the quality of chips is increasingly demanding. During the packaging process of chips, bubble defects frequently occur. Excessive bubbles might compromise the airtightness of the circuit and the stability of the cover, hence decreasing the reliability of the products. Therefore, internal bubble inspection of chips is a crucial component of the manufacturing process, playing a crucial role in regulating chip quality. At present, the detection of bubble defects in chip X-ray images mainly relies on manual operation, which has low accuracy, low efficiency, and expensive labor costs. Therefore, accurately and effectively detecting bubble defects in chip X-ray images is crucial.

Earlier works on chip defect detection have focused on traditional image processing methods. They use hand-crafted features, such as geometric features [1], color features [2], texture features [3], scale-invariant characteristic transform (SIFT) [4], local binary patterns (LBPs) [5], and other features to detect defects. For example, Chen et al. [6] proposed an

inspection system for integrated circuit (IC) molding surfaces based on surface texture features, including normalization, shrinking, segmenting, and Fourier-based image restoration and defect identification, achieving a high accuracy rate of 94.2%. Lin et al. [7] proposed a region segmentation search-based defect detection method for complex structure chips, which incorporates the canny operator, contour interpolation, and shape interference elimination. Zhong et al. [8] proposed a three-stage technique for defect detection in flexible integrated circuit package substrates (FICSs), including image contrast enhancement, standard template acquisition, and featuring probability calculation, which outperforms existing methods in the time–accuracy tradeoff. However, on the one hand, hand-crafted features are selected heavily relying on rich expert experience and require a large number of parameters to be adjusted. On the other hand, hand-crafted features are sensitive to complex backgrounds, noise, and uneven illumination, resulting in poor robustness. In addition, X-ray chip images have some challenges in bubble defect detection: (1) The overall gray value of the X-ray chip image is low, and the contrast between defects and background is low. (2) There is considerable interference from noise and complex background structures. (3) The scale of defects varies greatly, with irregular shapes and more defects of a small dimension. Therefore, it is difficult for traditional methods to accurately detect defects of different scales and shapes.

In recent years, with the development of computer vision technology, object detection networks based on deep learning have been successfully used in the field of defect detection, including fabric [9], metallic [10,11], railway [12], optical components [13], and so on. For instance, He et al. [11] proposed an end-to-end defect detection network for steel plate defects by combining multi-level features, achieving promising results with an accuracy of 99.67% for defect classification and 82.3% mAP for defect detection, and also contributed a defect detection dataset NEU-DET for fine-tuning. Lin et al. [14] proposed a convolutional neural network combined with class activation mapping for detecting light-emitting diode (LED) chip defects, reaching a classification error of 5.04% while also contributing an LCD chip dataset. Luo et al. [15] proposed a decoupled two-stage object detection model for FPCB surface defect detection, creating two distinct features for the localization and classification tasks, reaching an mAP of 94.15%, and built an FPBC surface defect detection dataset called FPCB-DET. Li et al. [16] proposed WearNet, a lightweight CNN, for surface scratch detection in contact sliding, achieving an impressive classification accuracy of 94.16% and successfully deploying the model on an embedded system, as well as gathering a large-scale surface scratch dataset. Object detection networks are mainly classified into two categories: one-stage and two-stage. Two-stage detection algorithms represented by R-CNN [17], Faster R-CNN [18], Cascade R-CNN [19], and Mask R-CNN [20] are divided into two steps. First, region proposals are created based on feature extraction, and then, these proposals are classified and regressed. One-stage detection algorithms represented by You Only Look Once (YOLO) series [21–24], Single-Shot Detector (SSD) [25], and RetinaNet [26] directly regress the classification and localization of targets based on feature extraction, which makes them meet real-time object detection tasks, but their accuracy is inferior to that of two-stage detection algorithms.

Currently, the YOLOv5 algorithm is frequently used for industrial defect detection because it has a considerable advantage in terms of detection accuracy and detection efficiency, resulting in major advancements in certain industrial circumstances. Zhang et al. [27] developed SOD-YOLO based on YOLOv5 for wind turbine blade (WTB) surface flaws, which improved the mAP by 7.82% when compared to YOLOv5. Du et al. [28] presented BV-YOLOv5S, a lightweight model designed for the detection of pavement defects. Compared to deep network learning models such as YOLOv3-Tiny, YOLOv5S, and B-YOLOv5S, BV-YOLOv5S demonstrated significant improvements of 4.1%, 3%, and 0.9%, respectively. Zhang et al. [29] proposed an enhanced YOLOv5-based solar cell defect detection method by incorporating deformable convolution and an ECA-Net attention mechanism, thereby increasing the mAP by 7.85% in comparison to YOLOv5. Shi et al. [30] proposed an improved YOLOv5 for the detection of steel surface defects by incorporat-

ing attention mechanisms and re-clustering anchor boxes, demonstrating a significant improvement in average precision, especially for small targets and targets with extreme aspect ratios. Wang et al. [31] suggested an improved MS-YOLOv5 model based on the YOLOv5 algorithm for detecting surface defects in aluminum profiles by using PE-Neck and multi-stream network components, reaching an AP of over 80% for each defect. The YOLOv7 algorithm [32], a YOLOv5-based enhanced network model proposed in 2022, surpasses existing target detectors on the MS COCO dataset in terms of speed and accuracy. In our practical implementations of chip bubble defect detection, YOLOv5 outperforms the state-of-the-art YOLOv7 model. Therefore, the YOLOv5 model is selected as the baseline. Nevertheless, there are several obstacles to overcome when applying YOLOv5 directly to the issue of bubble defect identification, including tiny target detection, imbalanced samples, and limited defective samples. Considering the above challenges, we develop a high-precision model, YOLO-Xray, based on YOLOv5 for chip X-ray image analysis. The main contributions of the paper are as follows:

1. We construct a dataset (CXray) as the research basis for bubble defects detection by foreground segmentation and homography transformation.
2. Various optimizations, which include the K-means++ clustering algorithm, the incorporation of micro-scale detection heads, the integration of the idea of BiFPN, and QFocal Loss function are introduced to improve the performance of the original YOLOv5 model.
3. Numerous experiments on the CXray dataset have verified the effectiveness of our proposed YOLO-Xray model.

The remainder of the article is organized as follows: Section 2 describes the acquisition and preprocessing of our CXray dataset. Section 3 presents detailed information on the improved YOLOv5. Section 4 elaborates on the experimental results and discussions. Section 5 draws conclusions and discusses future works.

2. Dataset

2.1. Data Acquisition

Nowadays, X-ray technology has been commonly used in nondestructive testing applications for chip package detection. Because X-rays can penetrate the chips to visualize the inner defects that are invisible to the naked eye without damaging them. X-ray images of chips used in this study were generated by X-ray inspection equipment from a chip testing factory, as shown in Figure 1. In this study, a typical type of chip was used as the research object. Additionally, the defect that this study focuses on the most is bubbles, which are typically seen in X-ray pictures of chips.

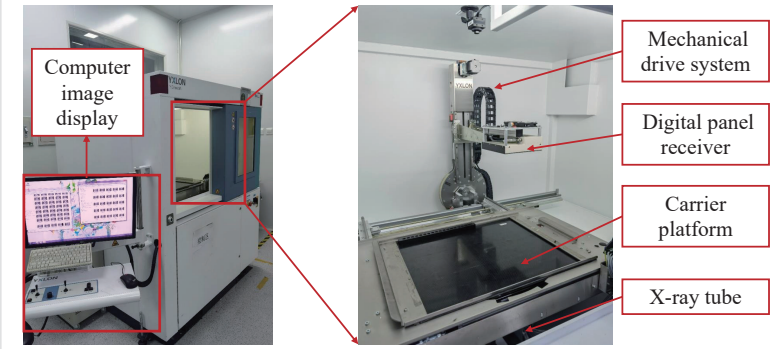


Figure 1. X-ray image inspection equipment.

2.2. Data Preprocessing and Annotation

As shown in Figure 2a, the original resolution of chip X-ray images collected by X-ray inspection equipment is 1004 × 620 pixels. However, the original image contains multiple

chips, and only the chip in the image's central region must be identified, as illustrated in the red-boxed part of Figure 2a. Since the chips may be placed obliquely, they are tilted in the image. Therefore, we develop a combination of foreground segmentation and homography transformation to extract and normalize the central chip.

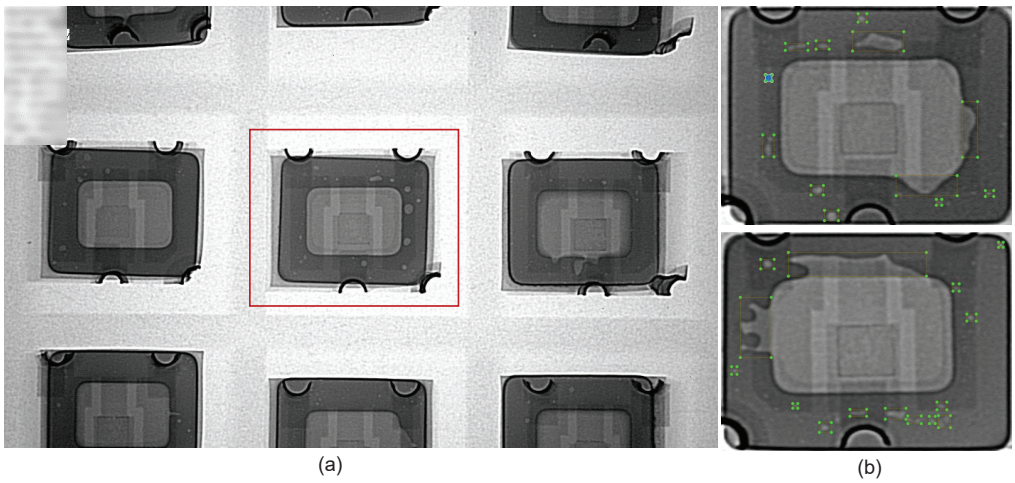


Figure 2. (**a**) Raw images captured by X-ray inspection equipment. (**b**) Samples of X-ray images with annotations of bubble defects.

The steps of the algorithm are described as follows:

1. The original images are center-cropped to 512×512 pixels.
2. Gaussian filtering of a kernel size of 5×5 is employed to remove Gaussian noise in the X-ray image, such as particle noise on the fluorescent screen.
3. The OTSU algorithm [33] is adopted to determine the threshold value for binarization segmentation.
4. Morphological opening and closing operations are applied to the image after segmentation for noise points filtering and hole filling.
5. Two-pass algorithms are used to label the contours of all connected regions and filter out the central chip based on the prior geometrical information.
6. The Homography transformation matrix, H, is applied to map an image from different scales to a uniform scale. Suppose that (x, y) and (x', y') are the pixel coordinates of the original image and transformed image, respectively. The form of transformation is shown in the following equation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

where $h_1, \ldots, h_8$ are transformation parameters. The H matrix can be solved through four feature points at least. We first solve the minimum area bounding rectangle (MABR) of the target contour based on an approximation algorithm. Then, we select four corner points of MABR as reference points(u, v), and (u', v') is the corresponding coordinate of four static boundary points of the ideal image.

After image preprocessing, we obtained an image containing only a single chip. We randomly selected 1000 images from real industrial environments with varying noise situations and lighting attributes. All images were annotated carefully into the VOC dataset [34] format by professionals using the labeling software called LabelImg. At the same time, the class names and coordinates of the bounding box of each bubble defect are stored in corresponding XML files. The CXray dataset holds 1000 images with around 10,000 bubble defects. The dataset is randomly split into the training set, the validation set,

and the test set, according to a ratio of 7:2:1. Some samples of the CXray dataset are shown in Figure 2b.

3. Method

3.1. YOLO-Xray Algorithm

In the previous section, we constructed a dataset of chip bubble defect X-ray images called CXray. The following are some of the characteristics of the CXray dataset: (1) The defects may appear at any location in the image. (2) There may be a large number of tiny defects exhibiting modest and dense distribution features. (3) There are large-scale long strips and irregular bubble defects of varying sizes, as shown in Figure 2b.

As one of the most advanced single-stage object detectors, the YOLOv5 algorithm [35] possesses advantageous characteristics such as rapid detection speed, high accuracy, and adaptability. However, when bubble defect identification is explicitly applied, there are still several obstacles to overcome. It has difficulty detecting small and irregular defects, resulting in a greater rate of missed detection.

To address the above problems, we propose an improved YOLOv5 model YOLO-Xray for detecting small defects and irregular defects in the CXray dataset. A schematic diagram of the structure of the YOLO-Xray model is shown in Figure 3, and the composition of each component is shown in Figure 4. The model is mainly composed of four parts: the input module, backbone network, feature extraction network, and detection network. In the input module, the K-means++ clustering algorithm was used to regenerate the optimal anchors suitable for our dataset. In the detection network, based on the original three detection heads, a micro-scale detection head was designed to improve the detection performance of small-sized targets. In the feature extraction network, the idea of bi-directional feature fusion was introduced to fuse feature information from multiple scales for better learning of irregularly shaped defect features. In the classification loss and confidence loss, the Quality Focal Loss function was used to replace the original cross-entropy loss function to solve the imbalance of positive and negative samples.

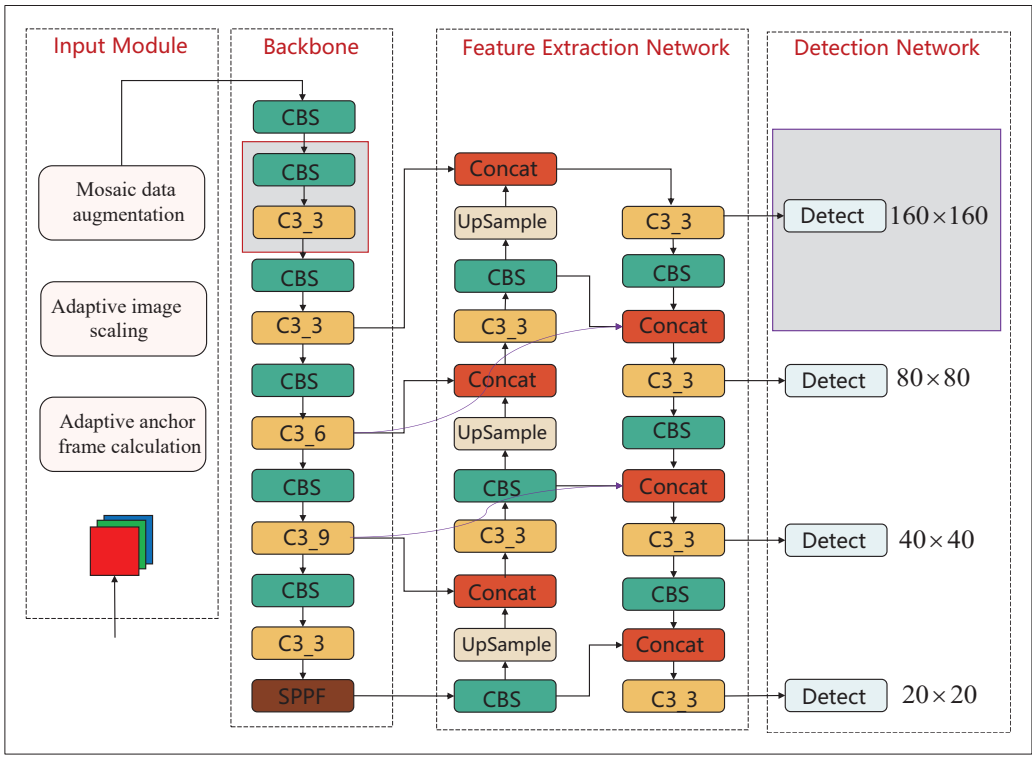


Figure 3. Schematic diagram of the structure of the YOLO-Xray model.

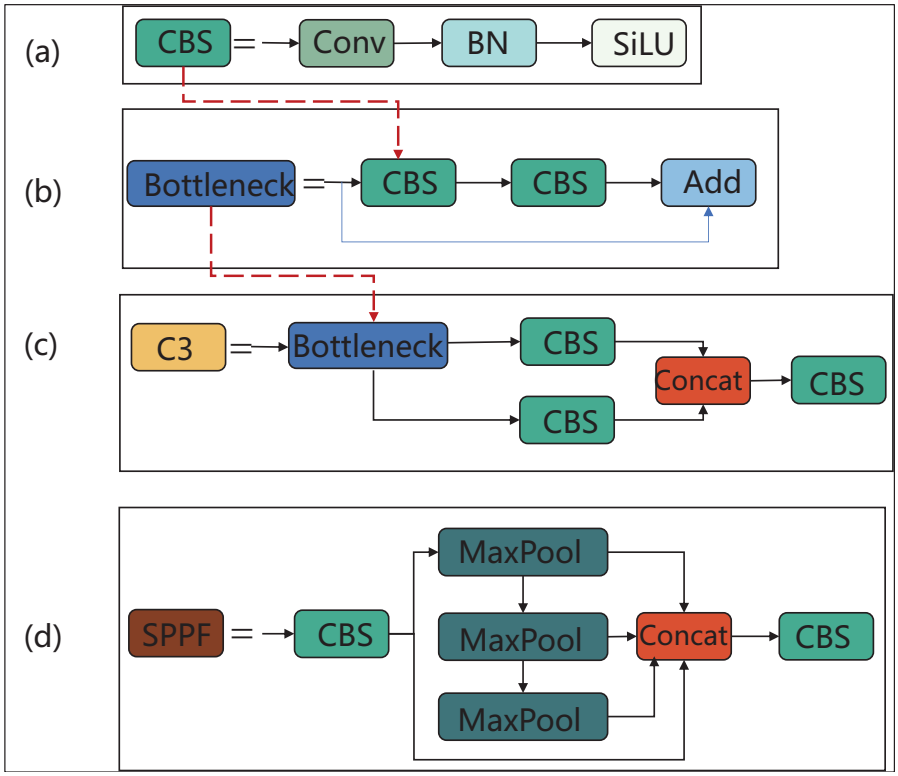


Figure 4. Schematic diagram of each functional module: (**a**) CBS; (**b**) bottleneck; (**c**) C3; (**d**) SPPF.

3.2. Anchor Design

The YOLOv5 model for object detection relies on the generation of a set of anchor boxes on the input image, with the selection of anchor boxes having a direct effect on the model's accuracy. In YOLOv5, when detecting a single category, predefined anchor boxes are utilized. However, the parameters of the predefined anchor boxes are derived from the COCO dataset [36], which has a large gap with the CXray dataset. Figure 5 depicts the size distribution of bubble flaws in the CXray dataset, revealing that our dataset contains a higher proportion of smaller-sized objects and larger aspect ratios. Considering that the CXray dataset focuses solely on detecting bubbles as defects, it becomes necessary to re-cluster the actual annotated bounding boxes.

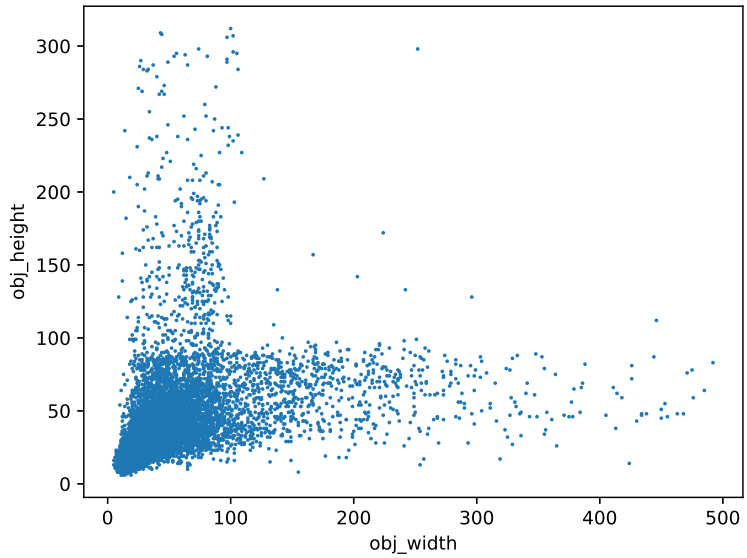


Figure 5. Size scatter distribution plots of bubble defects.

Therefore, the K-means++ algorithm [37] is used to re-cluster the prior anchor boxes that are suitable for our datasets. Table 1 displays the predefined anchor boxes and clustered anchor boxes.

The following is a description of how the K-means++ algorithm works in terms of its flow:

Table 1. Comparison of Anchor Boxes Before and After Improvement.

Feature Map Scale	Before	After
Large-scale	[116,90, 156,198, 373,326]	[38,34, 61,51, 142,57]
Medium-scale	[30,61, 62,45, 59,119]	[16,15, 21,19, 27,24]
Small-scale	[10,13, 16,30, 33,23]	[10,11, 14,12, 12,15]

Input: Read the CXray dataset and retrieve a set B of the widths and heights of all ground truth target boxes. Set K to the number of clustering centers.

Output: K anchor boxes.

1. Randomly select a value from the set, B, as the initial clustering center, C.
2. Calculate the minimum IoU distance, $d(x)$, between each sample, x, and the existing clustering centers, C. Determine the probability, denoted by $p(x)$, that each sample will serve as the subsequent clustering center. Select the next clustering center using the roulette wheel method, where samples with greater probabilities have a greater chance of being chosen.
3. Repeat step 2 until there are K clustering centers.
4. Calculate the IOU distance to the K clustering centers for each sample in the dataset and assign it to the cluster corresponding to the clustering center with the shortest distance.
5. Calculate the average width and height of each cluster and use those values as the new K clustering centers.
6. Repeat steps 4 and 5 until there is no longer any change in the positions of the clustering centers, resulting in the desired anchor boxes.

$d(x)$ and $p(x)$are defined as follows:

$$d(x) = 1 - IoU(x, C) \tag{2}$$

$$p(x) = \frac{d(x)^2}{\sum_{x \in B} d(x)^2} \tag{3}$$

where $x = (w1, h1)$, $C = (w2, h2)$, and IoU represents the intersection-over-union ratio between a sample point, x, and a clustering center, C, ranging from 0 to 1. If they are more similar, the IoU will be larger. IoU is defined as follows:

$$IoU = \frac{x \cap C}{x \cup C} \tag{4}$$

3.3. Micro-Scale Detection Head

The original YOLOv5 algorithm detects targets of different sizes using 8×, 16×, and 32× downsampled feature maps as feature layers. Assuming 640 × 640 image data as input, the three output detection scales of size 20 × 20, 40 × 40, and 80 × 80 were utilized to identify targets of large, medium, and small sizes, respectively. However, the CXRay dataset contains a large number of defects that represent only a small portion of the whole image. The model is required to have a strong ability for detecting the small object. While the original YOLOv5 primarily employs the feature layer after 8× downsampling to detect the small target, it neglected to utilize the target location information in the lower feature layers, resulting in the loss of the small target location. To solve the problem, we added a micro-scale small object detector compared with YOLOv5. The improved backbone network and detection network structure is shown in Figure 6.

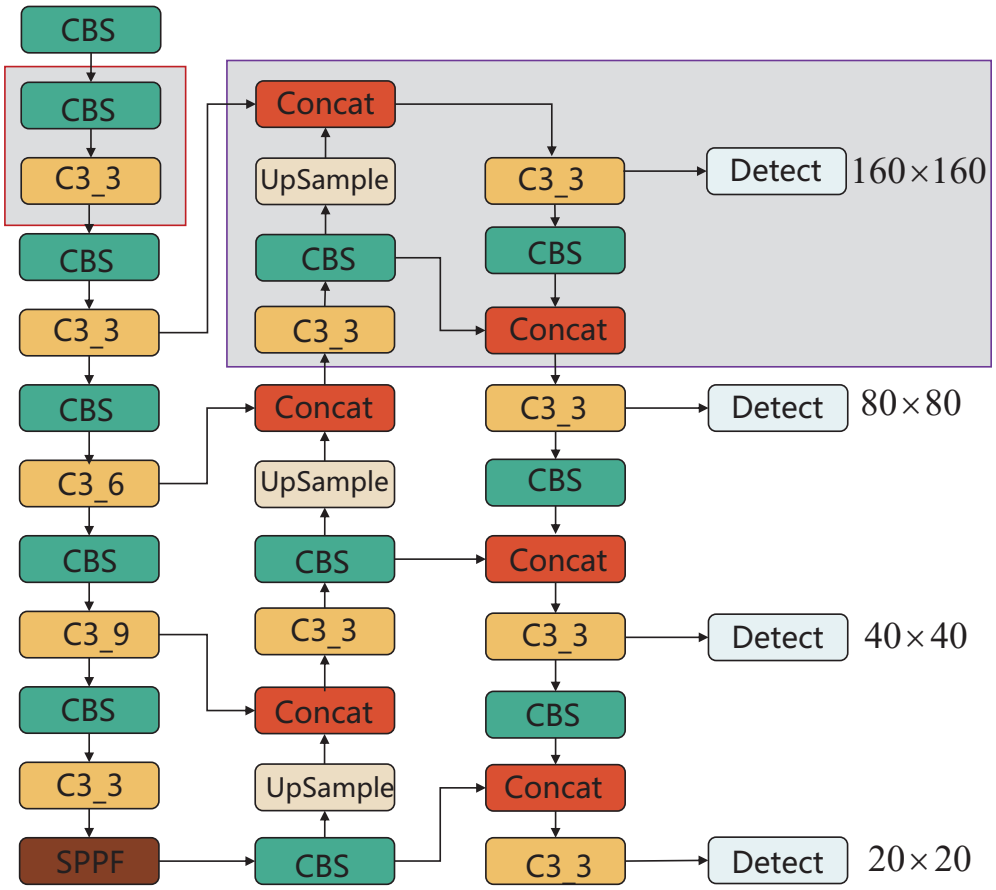


Figure 6. The structure of improved backbone and detection network.

Instead of using a 4× downsampling feature layer for feature fusion based on the original backbone network, we inserted a module (as depicted by the red box in Figure 6) between the 2× and 4× downsampling feature layers to improve the feature extraction capability and make better use of the micro-scale feature layer. The fusion module continues to employ the PANet [38] and FPN [39] networks for the feature fusion of different layers. The final output size for the small target detection layer is 160 × 160 (as shown in the purple box in Figure 6).

3.4. Bi-Directional Feature Fusion Module

The shallow features focus more on details and location information, which is good for localization, while deep features have stronger semantic information, which is good for classification. In the neck module, the YOLOv5 adopts the FPN structure and PAN structure to combine features of different layers. The FPN structure adopts a top-down structure, which can maintain the deeper semantic information but does not conceal the usage of shallow location and detail information, as shown in Figure 7a. The PAN structure adds a bottom–top downsampling path to the FPN structure, which can maintain the deeper semantic information and shallower location information, as shown in Figure 7b. However, in the CXRay dataset, there are long strip and irregular bubble defects with a limited sample size, and the majority of bubble defects only account for a small percentage of the overall image. To address the above-mentioned characteristics of defects and improve the defect detection accuracy, we updated the four scaled feature layers of the backbone network and referred to the concept of bi-directional feature fusion to build a new YOLOv5 feature fusion network.

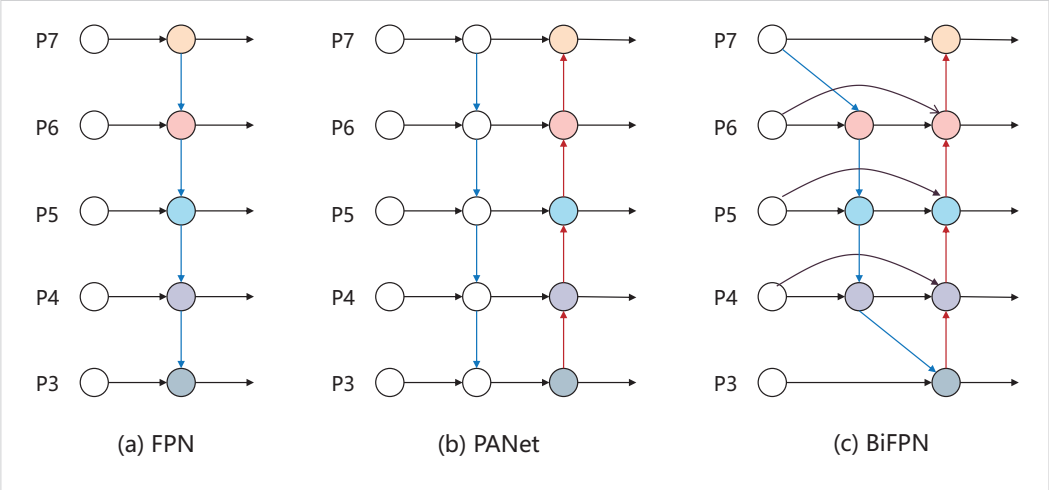


Figure 7. Structure diagrams of FPN, PANet, and BiFPN.

The BiFPN structure is a new feature fusion module suggested by Google for the EfficientDet network [40] that is based on PANet's architecture, as shown in Figure 7c. There are some significant design differences between the BiFPN and the PAN structure: (1) For the nodes with single input and output edges, because they are not involved in feature fusion, they contribute less to the network, so we removed the intermediate nodes of P3 and P7 in the PANet structure in Figure 7b. (2) Adding jump connections between input and output nodes at the same scale, as they are at the same layer, enables the fusion of more features without incurring an excessive computational expense.

Figure 8 illustrates the topology of the enhanced feature fusion module, in which the feature fusion technique is still merged by the number of channels. Feature layers acquired from the customized backbone network at 4×, 8×, 16×, and 32× downsampling are shown as $P_{2,1} - P_{5,1}$ in Figure 8. We eliminated the feature fusion layers $P_{2,2}$ and $P_{5,2}$ in layers 2 and 5, respectively, and introduced two jump connection lines that went from $P_{3,1}$ to $P_{3,3}$ and $P_{4,1}$ to $P_{4,3}$, respectively.

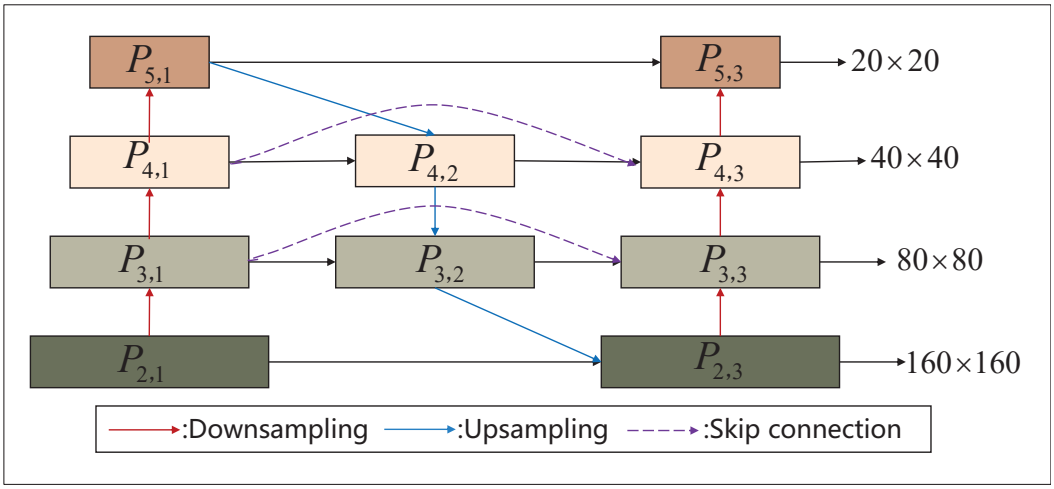


Figure 8. Improved feature extraction network.

3.5. Loss Function

The loss function of YOLOv5 consists of three parts: target confidence loss, localization loss, and category loss. The category loss and confidence loss are both binary cross entropy (BCE) losses, while the localization loss is the CIoU loss.

The BCE loss function is as follows:

$$L_{BCE}(y, \hat{p}) = -ylog(\hat{p}) - (1-y)log(1-\hat{p}) \tag{5}$$

where $\hat{p}$ is the probability of the predicted sample, and y is the true label.

Nevertheless, when detecting the CXray dataset, the model generates a large number of candidate frames, but these candidate frames contain fewer regions of the target (positive samples), and the majority of them are backgrounds that do not contain the target (negative samples), which can lead to the problem of positive and negative sample imbalance. The negative samples cannot be used to train a network, and a significant quantity of simple negative examples will degrade the model's performance during training, preventing the network from acquiring relevant information to effectively identify the target. In addition to positive and negative samples, there are also difficult and simple samples to consider. To solve the above problems, He et al. [26] suggested the focal loss for object detection networks, and this loss function may provide weight to each sample according to its predicted probability. The expression is as follows:

$$L_{focal} = \begin{cases} -\alpha(1-\hat{p})^{\gamma} log(\hat{p}), & \text{if } y = 1; \\ -(1-\alpha)\hat{p}^{\gamma} log(\hat{p}), & \text{if } y = 0. \end{cases} \tag{6}$$

where $\hat{p}$ is the probability of the predicted sample, y is the true label, α is the weight of a balanced positive and negative sample, and $(1-\hat{p})^{\gamma}$ and $\hat{p}^{\gamma}$ are the modulation coefficients.

However, focal loss only allows discrete category labels like 0 and 1, and confidence scores and classification scores are produced individually during the training process but combined during the inference step. Due to increased crossover and overlap between candidate frames when the targets are crowded, certain candidates with exact locations but low confidence are suppressed, which, in turn, affects the final detection results. To solve the above problems, Quality Focal Loss (QFocal Loss) [41] was used to replace the BCE Loss as the confidence loss and category loss. QFocal Loss is an extended form of focal loss on continuous label values, which can effectively balance both positive and negative as well as difficult and easy samples, and can also be adapted to the IOU-based supervision of continuous probability distributions. The expression is as follows:

$$L_{Qfocal} = -\alpha_t * |y-p|^{\beta}[(1-y)log(1-p) + ylog(p)] \tag{7}$$

where $\hat{p}$ is the probability of the predicted sample, y is the 0∼1 label after smoothing, $\alpha_t = y * \alpha + (1-y) * (1-\alpha)$ is used to balance positive and negative samples, and $|y-p|^{\beta}$ is used to balance difficult and easy samples.

In this paper, we keep the original CIoU loss adopted by the YOLOv5 as the bounding box loss function. The formula of the CIoU loss function is shown below:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{C^2} + \alpha v \tag{8}$$

$$\alpha = \frac{v}{(1-IoU)+v} \tag{9}$$

$$v = \frac{4}{pi}(artan\frac{w^{gt}}{h^{gt}} - artan\frac{w}{h})^2 \tag{10}$$

where ρ^2 represents the Euclidean distance between the prediction box's center point and the ground truth box's center point. c is the diagonal length of the minimal enclosing box that encompasses both the prediction box and the ground truth box. w^{gt} and h^{gt} represent the width and height of the ground truth box, respectively, and w and h represent the width and height of the prediction box, respectively.

4. Experiments and Discussion

In this part, we first introduce the implementation detail and evaluation criterion. Then, we design two sets of experiments on the CXray dataset, the first of which conducts the ablation experiments, and the other compares the performance of the improved YOLO-Xray with the mainstream one-stage and two-stage object detection models. Finally, we visualize the qualitative results of chip determination.

4.1. Implementation

We conducted bubble defect detection experiments using the CXray dataset. The configuration of the experimental environment is described as follows: Ubuntu18.04, 128 G RAM, Intel Core i9-10920X, and NVIDIA RTX 3090Ti with 24 G memory. The model training and testing are conducted under PyTorch1.8.0 and CUDA11.1. The training hyperparameters are as follows: the input size was 640 × 640, the training epoch was set to 300, the batch size was set to 16, and the learning rate was set to 0.001.

4.2. Evaluation Criterion

Precision (P), recall (R), and $F1$-score are commonly used as evaluation metrics to assess the performance of the model. Precision is the model's capacity to precisely identify the detected objects. Recall denotes the model's capacity to identify and capture all instances of the target objects. $F1$-score takes into consideration both precision and recall and can be viewed as their harmonic mean. They are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$R = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{13}$$

where TP is the number of bubble defects that are correctly detected; FN is the number of bubble defects that are not detected; and FP is the number of incorrectly bubble defects. The mAP is the mean value of the AP of all classes.

The precision–recall (P–R) curve depicts the relationship between precision and recall graphically. Precision is represented on the vertical axis (y-axis), and recall is plotted on the horizontal axis (x-axis) in the P–R curve. The average precision (AP) is defined as the area under the P–R curve. The mean average precision (mAP) represents the average accuracy across all classes. These are defined as follows:

$$AP = \int_0^1 P(R)dR \tag{14}$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{15}$$

where n is the number of classes.

In engineering practical applications, the speed of detection and the size of the model are crucial factors. The frames per second (FPS) metric is used to evaluate the speed of the model during the testing stage. The model size is the quantity of storage space needed to store the model.

Therefore, the mean average precision (mAP), the frames per second (FPS), and the model size are used as metrics to assess the detection performance of the experiments.

4.3. Ablation Experiments

To verify the effectiveness of the proposed different improvements on the performance of the previous YOLOv5 algorithm, the five ablation studies were designed to evaluate

the impact of K-means++, micro-scale detection layer, BiFPN, and QFocal Loss. A detailed comparison of different configurations is listed in Table 2, where ✓and ✕ represent the corresponding improvement strategies used and not used in the experiments, respectively.

Table 2. Experimental Results of Different Improvements.

Method	K-Means++	Micro-Scale Head	BiFPN	QFocal Loss	mAP0.5	Detection Speed (FPS)	Model Size (Mb)
A (the original YOLOv5)	✕	✕	✕	✕	0.884	**156.25**	**6.70**
B (A+ K-means++)	✓	✕	✕	✕	0.905	156.25	6.70
C (B+ micro-scale head)	✓	✓	✕	✕	0.912	138.89	8.34
D (C+ BiFPN)	✓	✓	✓	✕	0.919	119.05	9.56
E (YOLO-Xray)	✓	✓	✓	✓	**0.935**	119.05	9.56

The values in bold show the best results.

From Table 2, it can be observed that the baseline (original YOLOv5) achieved an mAP0.5 of 88.4% and the detection speed was 156.25 FPS.

A → B: Method B was to use the K-means++ technique to establish the priori boxes for the original three detection heads before the training step, with the same amount of prior frames as in the baseline model, resulting in no increase in model size or inference time. Because the updated priori boxes were better suited to the goal size of the dataset, method B outperformed the baseline model by 2.1% while having no influence on model size or detection speed, demonstrating the efficiency of the K-means++ clustering algorithm.

B → C: Method C was used to better extract the feature of tiny targets, and a micro-scale target detection head was introduced to the backbone of YOLOv5. The feature fusion still uses a combination of FPN and PANet to output a new detection head for tiny target prediction. As a consequence, the upgraded model was able to detect more tiny targets that were previously difficult to detect, boosting the mAP by 0.7%. However, because the new detection head increased the number of parameters, the detection speed was slower than previously.

C → D: Method D introduced the bi-directional feature fusion concept of BiFPN, which provides efficient multi-scale feature fusion via cross-level connections and sibling jump connections. Method D enhanced the detection of irregularly shaped and dense objects, the mAP was increased by 0.7%, and the detection speed was 119.05 FPS.

D → E: Method E used QFocal Loss as the loss function to fix the problem of an uneven number of positive and negative samples, as well as hard and easy ones, and the mAP increased by 1.6%. The loss function was optimized during the training phase, without changing the model size or detection speed. Finally, the mAP of the improved YOLO-Xray model reached 93.6%, which is 5.1% higher than the baseline model, proving the effectiveness of the improvement. When compared to the original YOLO5, the detection speed of YOLO-Xray is 37.20 FPS slower, yet it can still detect bubble defects in real time.

The results demonstrate the effectiveness of our improvements in the following four stages: input module, backbone network, neck module, and loss function. The improved YOLO-Xray can accurately detect bubble defects, including tiny targets and incomplete targets.

4.4. Performance Comparison of Different Object Detection Algorithms

To further evaluate the detection performance of the improved YOLO-Xray proposed in this study, the algorithm was compared with mainstream object detection models, which include two-stage networks (Faster R-CNN [18]) and one-stage networks (EfficientDet [40],

YOLOv4 [24], YOLOv5 [35], and the latest YOLOv7 [32]) under the same environment configuration. All the models were trained and tested on the CXray dataset. The mAP and FPS were used to evaluate and compare each detection algorithm, and a comparison of the experimental results are shown in Table 3.

Table 3. Performance Comparison between Mainstream Detection Models.

Model	mAP0.5	Detection Speed (FPS)
Faster R-CNN	0.907	30
EfficientDet	0.845	73
YOLOv4	0.865	57
YOLOv5	0.884	156
YOLOv7	0.867	122
YOLO-XRay	**0.935**	**119**

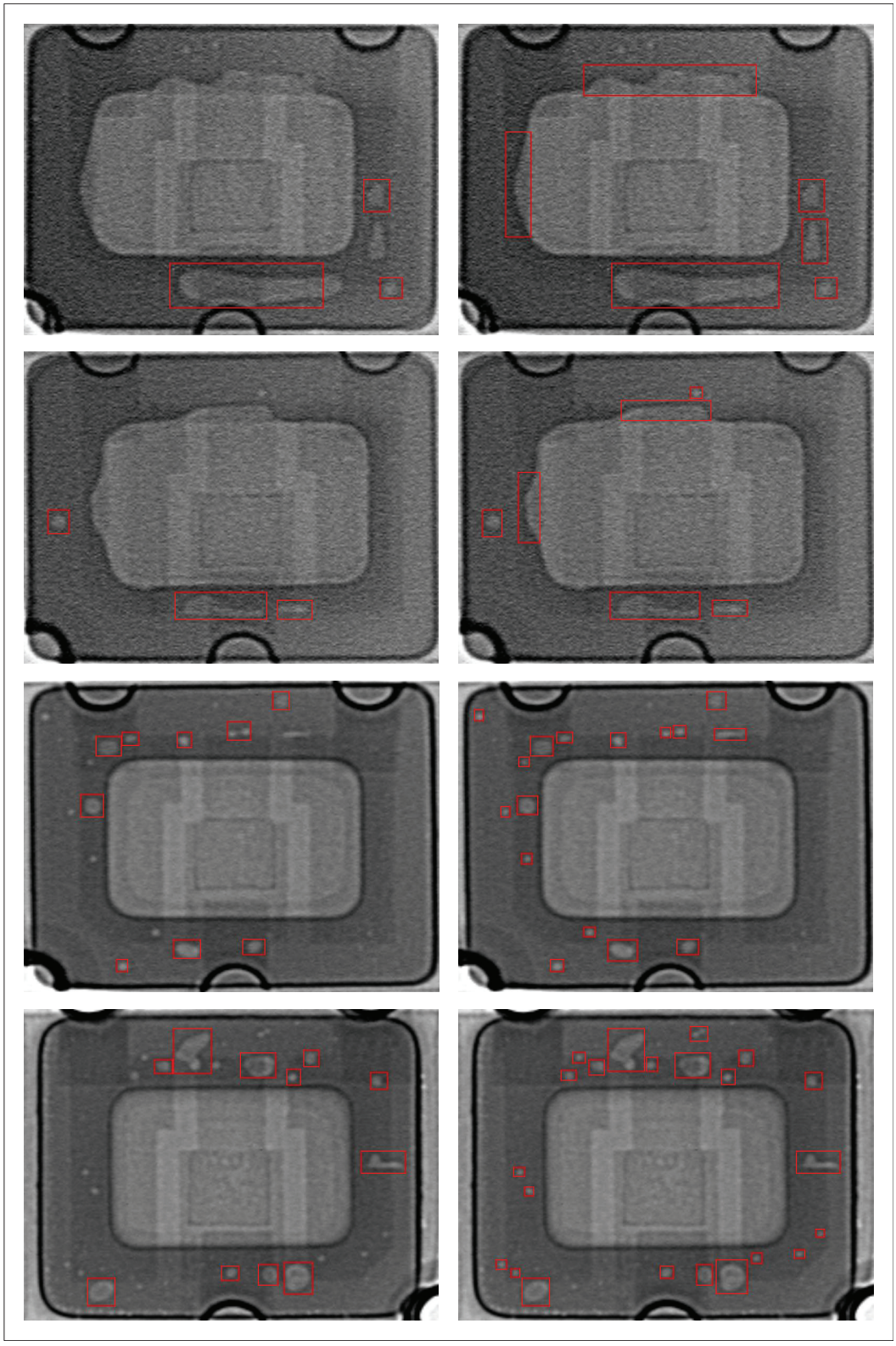
The values in bold show the best results.

According to Table 3, it can be seen that among the detection results, our YOLO-Xray model is the best in terms of overall performance. Although YOLOv7 outperformed many object detectors on the MS COCO dataset, with an map50 of 69.7%, the detection performance on the CXray dataset was inferior to that of the original YOLOv5. Compared to YOLOv7, our YOLO-Xray has higher detection accuracy, and the detection speed is close to that of YOLOv7. Faster R-CNN has a better detection accuracy of over 90% mAP, but the worst detection speed at only 30 FPS. The YOLO-Xray achieved 93.5% mAP, 2.8% higher than Faster R-CNN, 9% higher than EfficientDet, 7% higher than YOLOV4, 5.1% higher than YOLOV5, and 6.8% higher than YOLOv7. At the same time, the efficiency of YOLO-Xray is much better than other models, reaching 119 FPS, which is 4, 1.6, and 2.1 times faster than Faster R-CNN, EfficientDet, and YOLOV4, respectively. Based on the extensive analysis presented above, our suggested YOLO-Xray model has the greatest performance in terms of detection accuracy and speed for bubble defect detection, demonstrating the efficacy of our model.

4.5. Visualization of Defect Detection Results

Figure 9 illustrates some detection results of the original YOLOv5 and YOLO-Xray for the CXray dataset under various noise and contrast circumstances.

By comparing Figure 9, it can be found that in the first and second lines, there are some irregular bubble defects connected to the contour of the inner frame. The original YOLOv5 suffers from missed detection. While the improved YOLO-Xray accurately detects these bubble defects with high confidence. In the third and fourth lines of the detection results, there are several small and inconspicuous bubble defects. The original network is not ideal for detecting such bubble defects, and the leakage problem is serious. The improved model enhances the capability of detecting the types of defects and improves the detection of tiny bubbles. The results demonstrate the strong robustness and effectiveness of our YOLO-Xray model in complex environments.



(**a**) (**b**)

Figure 9. Comparison of detection results between the original YOLOv5 and the improved YOLO-Xray. (**a**) YOLOv5, (**b**) YOLO-Xray.

5. Conclusions

In this paper, we proposed a bubble defect detection model based on YOLOv5 to address the problem of poor accuracy and low efficiency of traditional methods. Firstly,

we constructed a dataset called CXray as the research basis for bubble defect detection, which is freely downloadable for research purposes. The dataset contains 1000 images with around 10,000 annotations. Various optimizations, which include the K-means++ clustering algorithm, the incorporation of micro-scale detection heads, BiFPN, and QFocal loss, were introduced to improve the performance of the original YOLOv5 model, which can effectively detect small and irregular bubble defects. The experimental results show that the improved YOLO-Xray increased the mAP from 88.4% to 93.5% compared to the original YOLOv5 and outperformed mainstream object detection models in terms of detection accuracy and speed on the CXray dataset. For chip manufacturers, our proposed YOLO-Xray model offers substantial benefits. It can considerably enhance the precision of defect detection, resulting in higher product quality and lower failure rates. In turn, this serves to reduce the costs associated with manufacturing defects. In addition, YOLO-Xray's high frames per second (FPS) performance reduces inspection time, resulting in increased efficiency and meeting the industry's detection requirements.

However, there are some future research areas for improvements:

1. Training the YOLO-Xray model requires a large quantity of annotated datasets, which can be both costly and time-consuming to acquire. Future research should investigate data augmentation techniques and the application of generative adversarial networks to generate synthetic defect samples. These methods can reduce the need for manually annotated data and accelerate the training process.
2. The YOLO-Xray model is based on enhancements made to the original network. In light of the need for lightweight models, additional research should investigate strategies for lightweight network architectures that reduce the parameter size of the model. This would allow its deployment on embedded devices, thereby expanding its applicability.

Lastly, we will continue to optimize our YOLO-Xray model and explore its application to different product types, aiming to expand its scope and versatility.

Author Contributions: Conceptualization, X.Z. and J.W.; methodology, J.W.; software, J.W.; validation, J.W., B.L., G.L., Y.Z. and L.Z.; formal analysis, J.W.; investigation, J.W.; resources, X.L.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, J.W.; visualization, J.W.; supervision, X.Z.; project administration, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CXray dataset is open and available at https://github.com/EudicL/CXray (accessed on 12 July 2023).

Acknowledgments: The authors wish to acknowledge China Aerospace Components Engineering for providing the X-ray images of chips used in this study.

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

References

1. Choi, K.J.; Lee, Y.H.; Moon, J.W.; Park, C.K.; Harashima, F. Development of an Automatic Stencil Inspection System Using Modified Hough Transform and Fuzzy Logic. *IEEE Trans. Ind. Electron.* **2007**, *54*, 604–611. [CrossRef]
2. Peng, X.; Chen, Y.; Yu, W.; Zhou, Z.; Sun, G. An online defects inspection method for float glass fabrication based on machine vision. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 1180–1189. [CrossRef]
3. Chetverikov, D.; Hanbury, A. Finding defects in texture using regularity and local orientation. *Pattern Recognit.* **2002**, *35*, 2165–2180. [CrossRef]
4. Lowe, D. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Corfu, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157. [CrossRef]
5. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]

6. Chen, S.H.; Perng, D.B. Automatic optical inspection system for IC molding surface. *J. Intell. Manuf.* **2016**, *27*, 915–926. [CrossRef]
7. Lin, B.; Wang, J.; Yang, X.; Tang, Z.; Li, X.; Duan, C.; Zhang, X. Defect Contour Detection of Complex Structural Chips. *Math. Probl. Eng.* **2021**, *2021*, 5518675. [CrossRef]
8. Zhong, Z.; Ma, Z. A Novel Defect Detection Algorithm for Flexible Integrated Circuit Package Substrates. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2117–2126. [CrossRef]
9. Kumar, A. Computer-Vision-Based Fabric Defect Detection: A Survey. *IEEE Trans. Ind. Electron.* **2008**, *55*, 348–363. [CrossRef]
10. Ferguson, M.; Ak, R.; Lee, Y.T.T.; Law, K.H. Automatic localization of casting defects with convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 1726–1735. [CrossRef]
11. 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.* **2020**, *69*, 1493–1504. [CrossRef]
12. Chen, J.; Liu, Z.; Wang, H.; Núñez, A.; Han, Z. Automatic Defect Detection of Fasteners on the Catenary Support Device Using Deep Convolutional Neural Network. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 257–269. [CrossRef]
13. Tang, S.; He, F.; Huang, X.; Yang, J. Online PCB Defect Detector on a New PCB Defect Dataset. *arXiv* **2019**, arXiv:1902.06197.
14. Lin, H.; Li, B.; Wang, X.; Shu, Y.; Niu, S. Automated defect inspection of LED chip using deep convolutional neural network. *J. Intell. Manuf.* **2019**, *30*, 2525–2534. [CrossRef]
15. Luo, J.; Yang, Z.; Li, S.; Wu, Y. FPCB Surface Defect Detection: A Decoupled Two-Stage Object Detection Framework. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
16. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef]
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
19. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162. [CrossRef]
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
22. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
23. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
24. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
26. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [CrossRef]
27. Zhang, R.; Wen, C. SOD-YOLO: A Small Target Defect Detection Algorithm for Wind Turbine Blades Based on Improved YOLOv5. *Adv. Theory Simul.* **2022**, *5*, 2100631. [CrossRef]
28. Du, F.J.; Jiao, S.J. Improvement of lightweight convolutional neural network model based on YOLO algorithm and its research in pavement defect detection. *Sensors* **2022**, *22*, 3537. [CrossRef]
29. Zhang, M.; Yin, L. Solar cell surface defect detection based on improved YOLO v5. *IEEE Access* **2022**, *10*, 80804–80815. [CrossRef]
30. Shi, J.; Yang, J.; Zhang, Y. Research on steel surface defect detection based on YOLOv5 with attention mechanism. *Electronics* **2022**, *11*, 3735. [CrossRef]
31. Wang, T.; Su, J.; Xu, C.; Zhang, Y. An intelligent method for detecting surface defects in aluminium profiles based on the improved YOLOv5 algorithm. *Electronics* **2022**, *11*, 2304. [CrossRef]
32. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
33. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
34. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.M.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
35. Ultralytics. YOLOv5. 2022. Available online: https://github.com/ultralytics/yolov5 (accessed on 20 June 2023).
36. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision (ECCV), COMPUTER VISION–ECCV 2014, PT V, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Volume 8693, pp. 740–755. [CrossRef]

37. Vincent, O.; Makinde, A.; Salako, O.; Oluwafemi, O. A self-adaptive k-means classifier for business incentive in a fashion design environment. *Appl. Comput. Inform.* **2018**, *14*, 88–97. [CrossRef]
38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. *arXiv* **2018**, arXiv:1803.01534. [CrossRef]
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. [CrossRef]
40. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *arXiv* **2019**, arXiv:1911.09070. [CrossRef]
41. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *arXiv* **2020**, arXiv:2006.04388.